A Review of Architecture Tools for the Australian Defence Force

Paul Prekop, Gina Kingston,
Moira Chin and Anna McCarthy

DSTO-TR-1139

20010926 103

A Review of Architecture Tools for the Australian Defence Force

Paul Prekop, Gina Kingston, Moira Chin and Anna McCarthy

Joint Systems Branch
Electronics and Surveillance Research Laboratory

DSTO-TR-1139

ABSTRACT

Complex defence architecture development efforts require the support of sophisticated enterprise architecture tools. This report identifies over 20 different enterprise architecture tools, and reviews four representative tools in detail. Several alternative approaches are described, including the current CAT development effort.

RELEASE LIMITATION

Approved for public release

AQ FOI-12-2584

A Review of Architecture Tools for the Australian Defence Force

Executive Summary

Complex defence architecture development efforts require the support of sophisticated enterprise architecture tools. This report identified over 20 different tools ranging from re-positioned CASE tools; expanded Business Process Re-engineering based tools, and general purpose enterprise architecture tools.

From this initial collection of 20 tools, four tools were selected for more detailed review; PTech Inc.'s Framework, Computas AS's METIS, Popkin Software's System Architect 2001 (SA 2001) and Proforma Corp's ProVision. These four tools were selected because they represent a fair sample of the type of architecture tools currently available.

Logicon Inc.'s JCAPS tool is also described, however, it was not reviewed in the same detail as the above four tools, because a demonstration version of JCAPS could not be made available to the review team.

The detailed reviews of the four tools were undertaken using a review framework that consisted of two dimensions, a functional dimension -- which reviewed how well the tools could support the architecture development activity, and a user communities dimension -- which reviewed how suitable the tools would be for various Australian Defence organisation user communities.

The following table summarises the results of the functional review of the four tools.

Functional Area	Framework	METIS	SA 2001	ProVision
Methodologies and Models	3/5	3/5	1/5	3/5
Model Development Interface	2/5	4/5	3/5	3/5
Tool Automation	2/5	1/5	3/5	2/5
Extendibility and Customisation	4/5	4/5	1/5	1/5
Analysis and Manipulation	3/5	3/5	1/5	Unknown
Repository	3/5	1/5	4/5[1]	4/5[2]
Deployment Architecture	1/5	1/5	4/5[1]	4/5[2]
Costs and Vendor Support	1/5	2/5	3/5	1/5[3]
1. For the multi-user version of System Architect.				
2. With the inclusion of the BOSS multi-user repository.				
3. This includes the BOSS multi-user repository. See review for additional information.				

The following table summarises the results of the user community's review of the four tools.

User Communities Dimension	Framework	METIS	SA 2001	ProVision
Capability Developers	3/5	3/5	2/5	2/5
Operational Planners	3/5	3/5	1/5	2/5
Architectural Researchers	4/5	4/5	1/5	1/5

Finally, this report presents some alternative approaches Defence may wish to consider, including the development of a custom tool, and the integration of COTS tools. The current CAT development effort is described in this context.

Authors

Paul Prekop
Joint Systems Branch

Before joining DSTO, Paul was involved in CML/CAL research at Deakin University's School of Engineering and Technology. Since joining DSTO in 1997, Paul has been involved in a variety of research fields, most recent of which involved modelling ADF operations using the C4ISR Architecture Framework.

Gina Kingston
Joint Systems Branch

Gina was born in Tasmania. She has a BSc with first class Honours from the University of Tasmania and a PhD from the University of New South Wales. Gina has been at DSTO for over 10 years and she currently works for the Joint Systems Branch in Canberra. She likes to draw on information from diverse areas such as mathematics, software engineering, process modelling, measurement theory and experimentation.

Moira Chin
Joint Systems Branch

Moira Chin is an Executive Officer within Joint Systems Branch. She has a Bachelor of Science in Computer Science and Psychology from the Australian National University. Moira joined DSTO in 1994, and since that time has been involved in a wide variety of research, including IT support to Strategic Planning and Exercises and the Command & Control Support Study. Recently, Moira lead a team which explored and developed Operational Architectures for the ADF.

Anna McCarthy
Joint Systems Branch

Anna joined JSB in January 2001. She has a BA with a major in Psychology and Sociology (Social Research Methods) from the Australian National University and a Graduate Diploma in Computing from the University of Canberra.

Acknowledgements

Contents

1. Introduction

Architectures are an emerging approach for capturing complex knowledge about organisations and systems. Architectural approaches range from broad, enterprise focused approaches, through to approaches aimed at specific user communities. Within the defence community, the US DoD's C4ISR Architecture Framework (C4ISR AF)[1] is emerging as one method, among many, for capturing the knowledge of how a defence force can be organised for particular situations.

Important to adoption of an architectural approach is the availability of tools to support the development, storage, presentation and enhancement of architecture representations. As with architecture methodologies, architecture tools to support the architectural development process are still emerging[2].

This report presents a review of architecture tools that are relevant for emerging ADF architecture development efforts. Over 20 different architecture tools have been identified and documented in Appendix A. A representative sample of four tools was selected from this list for more detailed analysis. The Review Framework, described in Section 2, was used as the basis for this analysis. The process used to identify the documented architecture tools is described in Section 3 of this report, with Section 4 containing the detailed reviews of the four representative tools.

As well as reviewing existing architecture tools, this reports describes alternative approaches for architecture tool development. Existing tool development efforts, as well as alternative approaches are discussed in Section 5 of this report. The final section, Section 6, describes several alternatives that may be useful for supporting the Australian Defence Organisation's (ADO) architecture development efforts.

---

[1] C4ISR AWG. (1997). *C4ISR Architecture Framework Version 2.0*. December 1997. US DoD.
[2] More information on the current state of ADF architectures can be obtained from the Defence Architecture Office.

2. Review Framework

To consistently review the architecture tools described in this report, the review team derived a review framework. The review framework consists of two dimensions: the basic functionality of the tool, and the utility of the tool to different user groups within the ADO.

When reviewing a tool's basic functionality, the review team attempted to describe how well the tool performed the different functions needed for the architecture development activity. The tools basic functionality was examined in the following areas:

- Methodologies and Models;
- Model Development Interface;
- Tool Automation;
- Extendability and Customisation;
- Analysis and Manipulation;
- Repository;
- Deployment Architecture; and
- Costs and Vendor Support.

Each functional area is described in more detail Section 2.1.

The second dimension, the tool's utility to different user groups, captures the fitness for purpose of the tool, and describes how useful the tool would be to particular ADO user communities The user communities considered were:

- Capability Developers;
- Operational Planners; and
- Architectural Researchers.

Each ADO user community is described in more detail in Section 2.2.

2.1 Functionality Dimension

This dimension of the review framework attempts to capture how well the tool performs the core functions needed to support the architecture development activity. This dimension breaks the functionality of an architecture tool into eight key areas. Each of the four representative tools was reviewed against this framework.

2.1.1 Methodologies and Models

The most important feature of an architecture tool the methodologies and modelling the approaches it supports. The approaches the tool supports dictate the types of architectures the tool is capable of supporting, and to an extent, the type of analysis and manipulation functions the tool is capable of performing. As well as reviewing the methodologies and modelling approaches, this functional area also reviews how well, or how completely, the tool implements the methodologies and modelling approaches it claims to support.

For tools that are capable of supporting multiple methodologies and modelling approaches, this functional area also examines how well the different approaches are integrated. For example, when complementary methodologies and modelling approaches (for example process modelling and data modelling) are used, how well can the different approaches be used together in an overall architectural approach? When a tool supports competing approaches (for example two approaches to data modelling) how well can the data being modelled be moved between the different perspectives offered by the competing approaches?

2.1.2 Model Development Interface

The model development interface is the most obvious part of an architecture development tool. It is the interface used to design, build, maintain and often manipulate, the models that make up the architecture. Generally, models are built and maintained graphically, by manipulating icons and the connections between them. The tool's model development interface may also use textual interfaces to allow additional information to be appended to the graphical models.

The overall quality of the model development interface is an important characteristic of any architecture development tool. The interface must support the modelling activity well, for example by automating some of the drawing functions, by automatically laying out models, or by providing pick lists of alternative values at the appropriate places during the modelling activity. The model development interface must also be intelligently structured, make good use of limited screen space, be logical and consistent to use and navigate. The tool should ideally follow the graphical user interface conventions and guidelines that apply to its host operating system.

2.1.3 Tool Automation

Developing and populating architecture models is often the most time consuming part of the architecture development activity. By providing support for automating parts of the architecture development processes, a tool can help speed up the overall development activity.

A tool may support the creation of macros or scripts, to automate common functions or actions, or to group several functions together into one action. These may be used to automate parts of the model development activity. This feature is closely related to the tool's ability to be customised, which is described in the next section.

The tool may also provide the ability to automatically generate architecture models based on data held within the tool's repository, or have the ability to generate architecture models as a result of data manipulation functions.

2.1.4 Extendability and Customisation

This functional group captures how well an architecture tool can be modified to meet the unique architectural requirements of the ADO. Architecture tools may support customisation by allowing users to add new modelling approaches or to modify the modelling approaches already supported by the tool. A tool may also support modification by providing a programming interface, allowing the functions of the tool to be modified, or allowing the tool to be integrated with other software products.

Most architecture tools that support high levels of customisation allow the underlying metamodels of the tool to be modified, and new metamodels added. Metamodels are literally models about models. They describe what entities can exist within particular models, the legal relationships between the different entities, and their properties. By modifying the existing metamodels, or adding completely new metamodels, a tool can be customised to support new modelling approaches.

The ability to modify the tool via a programming interface allows the functionality and behaviour of the tool to be customised to meet the unique requirements of the ADO. Programming customisation may be achieved though the use of an application scripting language, for example Visual Basics for Applications (VBA), or through support for adding external components, for example, Active X/DCOM components.

Architecture tools may be extended by integrating them with other software products. This may be achieved via direct integration through an exposed API within the tool, or via a middleware layer, for example ActiveX/DCOM, CORBA, and so on. Integration may also be supported via importing and exporting data into and out of the tool via standard file types; for example, character delimited or fixed width delimited text files, HTML, or SYLK files and so on.

2.1.5 Analysis and Manipulation

As well as supporting the development of architecture models, an architecture tool may also provide support for analysis and manipulation of the developed models. The type of analysis and manipulation support provided by the tool is often tied to the particular modelling approaches supported by the tool. For example, Flow Analysis is often tied to process/workflow modelling.

Analysis support provided by a tool may simply examine how correct or complete the model is, relative to a particular modelling approach used. More sophisticated analysis support may allow the model to be interrogated in some way, or be subjected to particular analysis methods. Analysis support may include the ability to compare different versions of models, allowing current and to-be architectures to be compared.

Manipulation functions capture a tool's ability to change the way the models are represented and viewed. This may include the ability to view models from particular perspectives, for example showing only particular classes of entities, or the ability to amalgamate separate models into a single model.

2.1.6 Repository

All the tools within this domain make use of some kind of data repository to hold the developed models. The functions provided by the tool's repository have a significant impact on the overall functionality, scalability and Extendability of an architecture tool. Some tools make use of commercial relational database management systems, or commercial Object Orientated or Object/Relational database systems, while others use proprietary repository systems.

A tool's repository often dictates the way users can collaborate. A repository may provide support for collaboration by supporting multiple, concurrent, users on the one repository, or by providing the ability to combine models developed by different modellers into one model.

The repository may also provide many different data management functions, including the ability to support model versioning, the ability to roll back to previous versions, the ability to lock parts of the model against change, and the ability to control access to part or all of the model.

2.1.7 Deployment Architecture

A tool's deployment architecture describes the tool's software structure and software implementation. Generally, architecture tools tend to adopt one of two deployment architectures: either a single user/single client structure, or a simple two-tier client/server structure.

Single user/single client structured tools are designed to operate on one workstation, and can generally only be used by one user at a time. Tools that implement this style of deployment architecture generally have a very tight coupling between the tool and its repository. In this type of deployment architecture, only one modeller can have access to the repository at any one time.

The second common deployment architecture found within the architecture tool domain is a simple two tier client/server structure. Tools that implement this style of deployment architecture generally have looser coupling between the tool and the repository. Generally, the repository is stored on a network server, and can often be accessed by multiple concurrent users. This deployment architecture allows multiple modellers to work on the same models concurrently.

2.1.8 Costs and Vendor Support

The final functional group considered is the cost of the tool and after sales support provided by the vendor. The cost of architecture tool licenses can range anywhere from $US 2,000 to $US 7,000 per license, and optional extras are often available for an additional cost. Given the high costs of this type of tool, the types of licensing agreements offered by the vendor, and how they may lower the overall cost, is important. For example, does the vendor support floating licences, allowing expensive licenses to be shared among a large group of users? Does the vendor offer discounts for bulk purchases, or site licences? Does the vendor offer discounts to government or defence organisations?

Also important in the overall cost of adopting an architecture tool, are the cost and type of maintenance and/or after sales support contracts offered by the vendor. Is the vendor able to offer comprehensive, in-house training? If the vendor is a foreign company, do they have an Australian representative available to provide training? Does the vendor offer free technical support? Is the vendor able to offer free or heavily discounted upgrades? How does the vendor address software faults discovered by the user? What are the yearly maintenance costs associated with the tool?

2.2 User Communities Dimension

The user communities dimension of the review framework captures the usefulness of the tools to Defence. Within Defence, three groups have been identified as being likely to gain significant benefits from the use of an architecture tool -- Capability Developers, Operational Planners and Architecture Researchers.

The evaluation of the tools considered their suitability for use by each of these user groups. The needs of other groups, such as software architects, are not considered in this report. These have been addressed in other papers[3].

The three groups in the ADO architecture community vary in their intent, their technical sophistication, and consequently their architectural tool requirements.

---

[3] Hunt J., *Mad About Modelling*. Application Development Advisor. May/June 1999.

2.2.1 Capability Developers

Capability developers include staff involved in the analysis and assessment of capability. For example, it includes staff in the Defence Information Environment Architectures Office and Joint Logisitics Support Agency. These users create enterprise architectures in order to make decisions about purchases, processes, doctrine and training. The architecture tools should be able to capture current and future resources (such as platforms, assets and components), organisations, people, information exchanges, tasks or activities, and processes and their relationships.

Capability developers need a tool that is easy to use, with support available when required. Local support is desirable, but probably not essential providing it is very responsive. The tool should have a strong drawing and analysis capability and allow reuse between architectures for different activities undertaken at different times. The ability to integrate with existing Defence data stores is very desirable.

2.2.2 Operational Planners

Operational planners, including HQAST and Strategic Command staff, create campaign architectures for specific campaigns or contingencies. Campaign architectures consist of a series of planned operational configurations that may be both time and event dependent. They need to be assembled and modified quickly, and should be based on current (or planned) Defence capability.

Operational planners need a tool that is easy to use. It is highly desirable that local (security cleared) support is available when required. The tool should have strong drawing and reuse facilities including support for multiple, related, configurations within a single architecture. Quick, automated, analysis and consistency checking is highly desirable. Integration with existing data sources is essential, particularly when implementing the plan.

2.2.3 Architectural Researchers

Architectural researchers investigate all aspects of architectural approaches and methodologies. This can involve researching different representations and architectural structures, including the development and investigation of alternative modelling approaches. For example, architecture researchers might investigate methods for adapting the C4ISR Architectural Framework (C4ISR AF) developed by the US DoD to support Australian requirements.

As such, the requirements for a tool to support architectural research are quite challenging. The over-arching requirement is flexibility in defining and adapting modelling approaches. However, a robust tool is also required to develop large-scale demonstrators to investigate, and promote these alternative approaches.

Unfortunately, flexibility can be both a benefit and a liability. The most flexible tools allow users to do anything they want, regardless of whether or not it is sensible. Thus, the needs of the architectural researchers may be at odds with those of other Defence users.

3. Tool Identification and Selection

The collection of potential architecture tools was obtained via comprehensive WWW search, as well as searching relevant technology portals (ZDNet and CMP's TechWeb). The vendors of the various tools were contacted, and additional information on each tool identified was obtained.

Currently, the architecture tool domain is broadly polarised into two classes of architecture tools: Extended CASE Tools, and Business Process Re-engineering (BPR) tools.

Many CASE tool vendors are positioning their products as being able to support the enterprise architecture development activity as well as being able to support the development of computer applications. While some CASE tools have considerable functionality to support the enterprise architecture development activity, over and above what may be required for the system development domain (which is the traditional domain of most CASE tools), most fall short and are still only suitable for providing support for the development of applications.

The second class of tools, often described as architecture or enterprise architecture tools, are Business Process Re-engineering (BPR) tools. BPR tools emerged during the late 1980s and early 1990s to provide tool support for BPR activities. Most tools within this class support some type of process modelling. Some of the more sophisticated BPR tools support other types of modelling approaches, for example, information flows, activities, organisational structures, and so on, as well as the ability to integrate the different approaches together.

During the initial WWW search for potential architecture tools, examples of both classes of tools were identified. However, only tools that supported one or more modelling approaches aimed at representing the structure and function of an enterprise, as well as enforcing the rules and constraints of the supported modelling approaches, were considered suitable as architecture tools. These tools are listed in Appendix A, and includes BPR tools and CASE tools, as well as several purpose-built enterprise architecture tools. While the list of tools found in Appendix A may not be complete, it can be considered a fair sample of the types of architecture tools currently available.

The view of architecture tools adopted by this report excludes *lower-CASE* CASE tools, and other specialist modelling, simulation tools and sophisticated drawing tools. Lower-CASE CASE tools tend to focus on modelling approaches and methodologies aimed at developing computer applications, rather than modelling approaches or methodologies aimed at modelling the enterprise. Most other specialist modelling and simulation tools suffer from similar limitations. Also excluded were most drawing tools. While drawing tools are capable of supporting almost any visual modelling

approach, they don't provide the ability to ensure the resulting models conform to the modelling approaches or methodology. They also tend to lack many of the model analysis and manipulation functions expected from an architecture tool.

From the list of potential architecture tools identified in Appendix A, a representative sample of four tools was selected for detailed review. When selecting these tools, the goal was to identify four tools that capture many of the features of the architecture tools identified in Appendix A. The four tools selected were subjected to a more detailed review process, described in Section 4.

PTech Inc.'s[4] Framework tool was selected because it not only supports many of the industry standard enterprise modelling approaches, but it also supports the C4ISR Architecture Framework. PTech's Framework is also highly customisable and extendable, and provides good model analysis functions.

Computas AS's[5] METIS is an interesting tool that is fully customisable. It only supports a few standard modelling approaches, but it provides a powerful metamodelling facility so end-users can add their own modelling approaches. METIS also includes a powerful view concept which allows the models to be manipulated, as well as including a sophisticated model development interface.

Popkin Software's[6] System Architect 2001 was selected as an example of a CASE tool that is emerging to provide support for enterprise modelling. System Architect 2001, supports some enterprise modelling approaches, as well supporting some levels of customisation. The tool also includes a process simulation tool, model analysis functions, and a multi-user repository.

Proforma Corp's[7] ProVision was selected as an excellent example of a comprehensive and slightly extendable BPR style tool. It provides support for a collection of methodologies and modelling approaches. ProVision also has an interesting way of implementing modelling approaches. Rather than supporting specific methodologies and modelling approaches, ProVision supports diagram types. ProVision also provides good model manipulation functions. The vendor is also currently reviewing the tool, and future releases of ProVision may include support for the C4ISR Architecture Framework.

One tool that could not be reviewed in any detail was Logicon Inc.'s JCAPS tool. Developed under contract from the US DoD, JCAPS is a tool designed specifically to provide support for the development and manipulation of C4ISR Architecture Framework products. At the time of writing, a version of JCAPS was not made

---

[4] PTech Inc.'s WWW Site: http://www.ptechinc.com/.
[5] Computas AS's WWW Site: http://www.metis.no/products/.
[6] Popkin Software's WWW Site: http://www.popkin.com.
[7] Proforma Corp's WWW Site: http://www.proformacorp.com.

available for review. However, a review of JCAPS, based only on the available JCAPS literature, is included in Section 4.6.

4. Tool Reviews

Each of the four tools identified in the previous sections were reviewed in detail by the review team. Evaluation copies of each tool were obtained from the vendors, although in the case of PTech Inc.'s Framework, copies already owned by DSTO were used. Each tool was reviewed against the review framework described in Section 2. In order to ensure consistency between the reviewers, an evaluation template was used to review each tool. A copy of the template can be found in Appendix B.

4.1 PTech Inc.'s Framework

PTech Inc. describes Framework as enabling organisations to *visually and logically capture their enterprise architectures*[8]. Framework has an enterprise wide focus, providing support for many enterprise and business modelling approaches, as well as providing modelling approaches to support application design and development. Framework is also one of the few tools to provide support for C4ISR Architecture Framework. It is customisable and extendable, making it capable of supporting new modelling approaches as they emerge.

Framework comes in three versions, Technology, Business and Enterprise. Technology Framework provides support for modelling approaches aimed at software applications development. Business Framework provides support for modelling approaches aimed at the business and enterprise functions and structures, for example, Business Architecture/Value Chain Models, Strategic Planning and Process/Workflow Models. Enterprise Framework includes the modelling approaches supported by Technology and Business Framework, as well as providing support for the Zachman Enterprise Framework. A customisation kit is also available for each version, which adds support for extending and customising Framework. Framework is only available for Microsoft's Windows 95, 2000, and NT 4.0.

The version of Framework reviewed in this report was the Enterprise Framework, Version 5.4.2., with the customisation kit. Framework exists as a self-contained, single user/single machine application. Installing framework is straightforward except for registering the application. PTech Inc. has taken an interesting approach to software security. To activate Framework, once it has been installed, a licence key is obtained from PTech Inc. This licence key only allows Framework to execute on the machine on which it has been installed. Moving the Framework to another machine, or upgrading/re-installing the machine's operating system, requires a new set of activation keys from PTech Inc.

---

[8] PTech Inc.'s WWW Site: http://www.ptechinc.com/.

Framework comes with a small set of printed manuals covering most of the basic functions of Framework. More advanced topics, for example customisation, queries or report writing, are covered in the electronic version of the manuals. The online help provided with Framework is sparse and important topics are often not covered. The printed manuals are a little better. Overall the documentation covers most of the basic topics well, however, the more advanced topics are poorly covered, or not addressed at all.

4.1.1 Supported Methodologies and Models

Framework supports a very wide collection of modelling approaches and methodologies, including: Business Objects, Rules, and Relationship Diagrams, Enterprise Activities/Process diagrams, enterprise requirements, and enterprise structure diagrams, as well as process diagrams, strategic planning, and team architectures diagrams. Framework also supports Forte Class and Event diagrams, and UML 1.1. Framework also supports class diagrams for JAVA, C++, as well as CORBA, and Oracle diagrams. With the addition of the PTech Inc.'s Military Information Architecture Accelerator (MIAA), Framework also provides some support for the C4ISR Architecture Framework, although it currently only supports the OV-1, OV-2, OV-4, SV-1 and SV-2 products.

Modelling in Framework is very flexible. While Framework does support formal modelling approaches, it is possible to combine elements of different modelling approaches together, in sensible, and sometimes not so sensible ways. This flexibility is a doubled edged sword. By allowing different modelling approaches to be combined, Framework is very flexible, and allows modellers to adopt the best modelling approaches to represent what they are trying to capture. However, this flexibility also means Framework will allow a modeller to develop models that don't actually conform to the modelling standard used. For example, it is possible within Framework to develop IDEF models that don't fully conform to the rules of IDEF.

Unlike some tools in this class, Framework doesn't rigorously enforce modelling approaches, nor does it provide any methods for checking a model's conformity to particular modelling approaches. Framework does ensure that all the entities and relationships in a model conform to the rules expressed in one or more metamodels. However, these metamodels need not come from the same modelling paradigm. Apart from providing little support for ensuring integrity of any models developed, Framework's implementation of the various modelling approaches it supports seems to be complete.

4.1.2 Model Developers Interface

Framework's Model Development Interface follows the almost standard layout for tools in this class. As shown by Figure 1 below, on the left of the screen is the navigation hierarchy. The navigation hierarchy allows collections of models,

metamodels, tool folders and queries to be organised in folders. The middle of the Framework panel is taken up by the model drawing area, and on the far right is the entity and relationship icon palette. Along the top are the drop-down menus and various toolbars.

On the bottom of the screen are two information windows -- the documentation window, and the status windows. The documentation window will show any text held within the documentation property of any modelling entity, or even a diagram. The status window shows the results of the various systems operations, for example, the results of running a report, or the results of a find operation. All the windows can be undocked from each other, repositioned, and shown or hidden independently of each other. It is useful to close both the documentation and status windows to save screen space.

Extensive use of tabs makes it easy to move between panels of the same type, for example moving between different model drawing area panels, or between different navigation views.
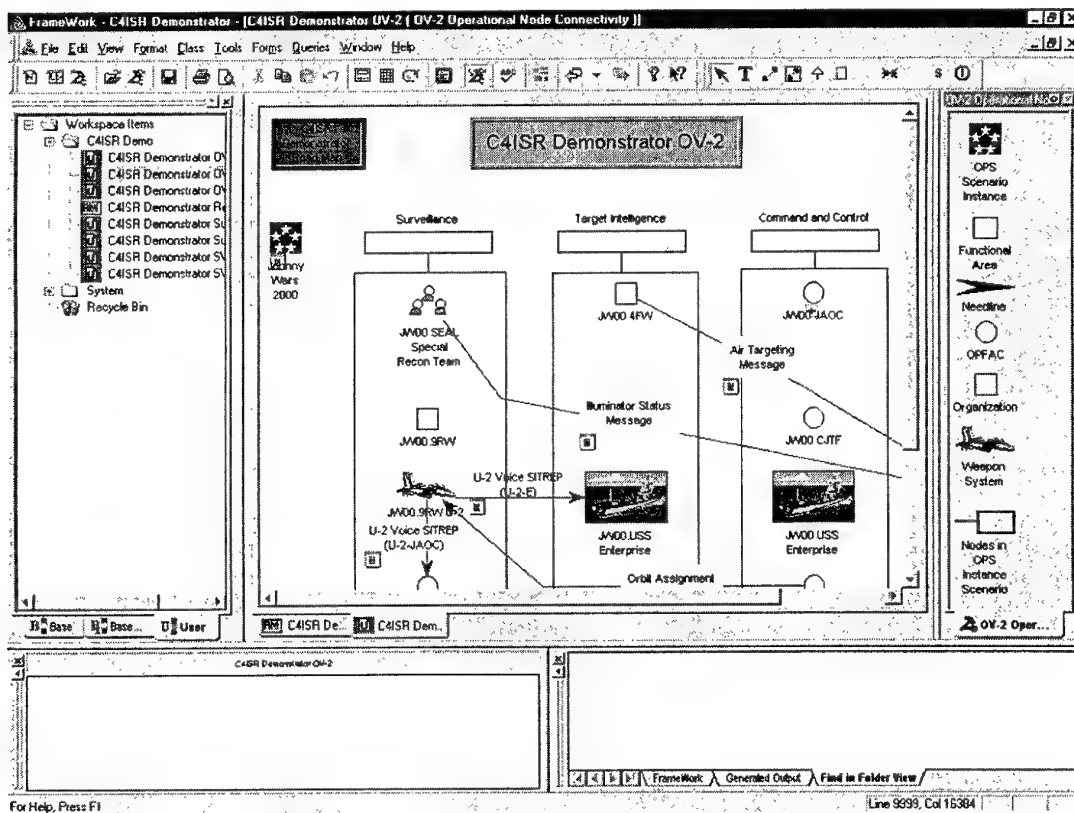

Figure 1 PTech Inc.'s Framework, Enterprise Edition, showing a C4ISR AF OV-2 model.

Framework's interface takes a little getting use to. Some of the drop-down menu options are hidden away in unexpected places, and there is often a strange division

between functions kept on the drop-down menus in the toolbars, and functions kept on the pop-up, contextual menus. However, overall, Framework follows most of Microsoft's Windows 95, 2000, and NT 4.0 GUI conventions.

Navigating through the collection of models can be achieved via the model navigation hierarchy or via hyper-links placed within the models themselves. The navigation hierarchy works as expected. Models can be held in different folders, and opened by clicking on them. The hyper-links, added by the model developer, make it easy to move from one model to the next simply by clicking on the hyper-links. By using hyper-links, it is possible to create various paths through a collection of models.

New models are created as a specific type of model. For example, as a C++ Class Model, or an Enterprise Activity model. Associated with each model type is the specific icon palette, called the tool folder in Framework. The icon palette holds the different objects and relationships that can be used in the modelling approach selected.

Models are developed by dragging icons representing objects, and icons representing relationships between the different objects, from the tool folder to the model drawing area. Only relationships associated with the relevant objects types in a metamodel can be used to link objects together. Framework's metamodel concept is described in more detail later in this review.

As well as using the relationships defined in the tool folder to link objects together, Framework also has a *box connect* tool, which allows a number of objects to be connected by dragging a rubber-band box around them and linking them to the relevant object. Framework also includes a *line connect* tool, which performs a similar function for individual objects. When using the box connect or line connect tools, the modeller can select the most relevant, legal association between the selected objects from a pop-up menu, rather than using the specific relationship that has been hard-coded into the relationship icon in the tool folder. While this is a powerful feature, and one that gives the modeller considerable flexibility in the development of models, the pop-up menus are often not wide enough to display the complete relationship name. Since the menu can't be horizontally scrolled, it difficult, sometimes impossible, to select the needed relationship from the pop-up menu.

Framework provides little support to aid the model drawing activity. Model developers must layout and organise their own models by hand. The only sophistication in the drawing interface is the support for bending lines at user selected anchor points, and anchoring the ends of lines to icons, so the lines move when the icon is moved.

While Framework allows the icons used to represent objects to be changed or new icons added, the process of changing or adding icon graphics is frustrating. Graphics added to Framework need to be in Windows Meta File (WMF) format or Windows Bit Map (BMP) format. Graphics are then dragged in to a special clip-art folder. Once in a

saved clip art folder, the graphics can be assigned to entities and relationships. The only place the graphic is associated with modelling entities or relationship is in the tool folder. This is fine when drawing a model, but if you want to change a graphic after the model has been developed, you need to change each instance of the entities and relationships. One annoying property of Framework is that clipart is stored in a single file, which is separate from any particular Knowledge Base. However, unless a clipart folder is exported and imported, it is only visible in the Knowledge Base in which it was created.

As well as representing models graphically, Framework also supports textual interfaces called Forms. Forms are simple containers for a small collection of widgets, including various list boxes, text boxes and tree controls, button and tab strip controls, and a data access control, which allows data in the Knowledge Base to be accessed. The form's display fields can be mapped to almost all modelling elements, including an object, the relationships between objects, and queries performed on an object. Forms can also be used to create new objects and relationships. Forms can add textual information to objects or relationships, and display the results of queries. As will be described later, forms can also be customised, and support very simple VBA scripts.

Overall, Framework's model drawing interface is a disappointment. It lacks the sophistication and features expected in tool of this class, and price range.

4.1.3 Automation, Customisation and Analysis

Framework provides support for tool automation through its forms. As discussed previously, forms are simple containers for a small collection of widgets, and can be seen as acting as a textual view of the model. Forms are associated with objects, relationships, or models. Forms in Framework include support for Visual Basic for Applications (VBA), and Active X components. Using VBA and the data access control provided with Framework, it is possible to build crude automation functions using forms. However, since forms can only be related to model entities and relationships, the scope of the functions that can be automated is limited. Also, since forms are designed for interactive use, the user must still interact with the form performing the automated function in some way; even if the user just needs to press the update button.

One of Framework's interesting and powerful features is its ability to be customised via its metamodelling facility. As discussed previously, metamodels are literally models about models. They describe how a particular modelling approach will be implemented, what objects exist within a model, and what relationships exist between the various objects. Framework allows end-users to add new metamodels, or modify and reuse part or all of the foundation metamodels supplied with Framework. While the foundation metamodels can be reused, they cannot be modified or deleted in anyway. This limits the utility of reusing the Framework foundation metamodels, since obsolete relationships and object types are still accessible.

Within Framework, developing metamodels is a straightforward process. Metamodels are developed in the same way as any other model. The metamodel elements in the form of objects and associations are dragged from the metamodel icon palette, and assembled as a complete metamodel. Once the metamodel has been developed, a unique icon palette (called a tool folder in Framework) can be built and associated with the metamodel. Building the icon palette is simple. The elements of the metamodel to appear on the icon palette, for example the various objects within the model, and the relationships between them, are dragged (with the Control key held down) from the metamodel diagram onto the new icon palette. Once added to the icon palette, the graphical properties of the elements can be set. For, example, the icon to represent the class can be modified, the style of line used in a relationship can be set, and so on. These graphical properties influence how the elements appear when used to develop specific models.

Once the metamodel, icon palette, and any relevant queries and forms have been developed, they collectively form a modelling approach in Framework, and from the user's perspective, they can be selected and used like any other modelling approach supported by Framework.

Framework provides the ability to lock user-developed models and metamodels into independent, read only-knowledge base segments. For example, the Military Information Architecture Accelerator (MIAA), which supports the development of C4ISR AF products, is implemented as a knowledge base segment. This segment can be added to any Framework knowledge base. Once added to the knowledge base, the modelling approaches held in the segment become available to the user of the knowledge base. End-users can define their own segments, making it simple to distribute support for new modelling approaches.

Framework provides some support for integration with other software applications, through its exporting and importing functions. Exporting data to other software applications can be performed by Framework's powerful *Template Language*. The template language is a stack based, flexible reporting language, which allows almost any data held in the knowledge base to be extracted, and written out to external files in any text-based structure. The language's syntax and structure is proprietary, and the documentation provided by PTech Inc. provides little more than a reference guide and some examples. The template language is extensively used by Framework to support its own code generation features.

Importing data *into* Framework is not as flexible. Data, in the form of delimited text files can be imported into Framework. However, the imported data must be structured to match the model into which it is being imported. The data is only imported into the knowledge base structures. As Framework doesn't create a graphical model of the imported data, this must be manually created by the user.

Framework is shipped with a collection of analysis methods. These analysis methods are tied to activity and process modelling, and resource modelling. For these modelling approaches, Framework supports Capability Analysis, Capability Measurement, Activity Based Costing, What-If analysis, and resource dependency assessments. The analysis functions provided by Framework are not particularly sophisticated. Generally, they involve adding considerable additional information into model structures, with the analysis performed by the user, scanning the structure for specific data, or lack of data. While the additional information is often needed to perform the types of analysis supported, the tool seems to provide little help in analysing the data in a meaningful way.

As well as the built-in analysis methods, Framework also has a powerful query and reporting language, both of which can be used to perform analysis on models. Framework's queries are based on a visual, set based, query language. They allow almost all the data held within the knowledge base to be accessed and manipulated. Queries are associated with particular modelling objects or relationships, which they use as the starting point for the query. The results of one query can be used as the input to the next query, used as the data for a field on a form, or displayed in a temporary window. Framework is supplied with a large collection of default generic queries, and users can add their own queries. Framework's template language, described previously, can also be used to perform analysis functions. The template language allows almost any element within the knowledge base to be accessed and manipulated, with results written to external files.

Framework provides only limited support for automated model manipulation. In Framework, it isn't possible to automatically merge models (while it is possible to import data into a model, which essentially merges two or more models together, the user is still responsible for generating the model diagrams), or to deconstruct models in any automated way. Framework does include the concept of hide levels, which allow selected objects and relationships to be hidden or shown, depending on the hide level the user elects. Hide levels need to be manually assigned to model elements, and it doesn't seem possible, for example, to assign a hide level to a particular meta-class.

4.1.4 Repository and Deployment Architecture

At the heart of Framework, as with most tools in this class, is the tool's repository. Within Framework, the repository is called the knowledge base. The knowledge base is a proprietary, single user OO style database, which is very tightly coupled to the Framework application. As a single user application, Framework provides no support for concurrent users, and PTech Inc., is not intending to release a Client/Server version of Framework within the short to medium term[9]. To facilitate collaboration within Framework, Framework's knowledge base supports the exporting and importing of

---

[9] Telephone Conversation with Larry John, Government Account Manager, PTech Inc. 22nd of April, 2001.

parts of the knowledge base to other Framework users. When importing models from one knowledge base to another, Framework performs simple deconfliction checking.

PTech Inc., will soon release an additional Framework product, the Knowledge Server/Portal. The Knowledge Server/Portal is scheduled for release during the fourth quarter of 2001. It is essentially a WWW server wrapper for the Framework Knowledge Base. It allows the data to be dynamically accessed and manipulated via a WWW browser. Initially only textual data will be able to be manipulated via the WWW browser[10].

4.1.5 Cost and Vendor Support

Framework is an expensive tool. Licences are around $US 7,000 per user. Site licences are available by negotiation, and discounts for bulk purchase can be arranged. Currently the Military Information Architecture Accelerator (MIAA) is available free when purchasing Framework, or it can be purchased as an independent tool for $US2,500 per licence. Some of this expense can be offset via the use of PTech's Licence Server. The Licence Server runs on a network server, and its role is to grant licences to, and revoke licences from, Framework sessions running anywhere on the network. This means, rather than having licences assigned permanently, floating licences are granted as needed. This makes it easier to spread the use of Framework licenses and to make a small number of licences available to a large number of casual users. Floating licences cost 1.5 times the cost of an fixed licence. PTech Inc. also offers support agreements at 15% of the licences' cost per year. The support agreements include free upgrades to the product, and basic phone and e-mail technical support.

As well as developing and marketing Framework, PTech Inc., also offers comprehensive architecture development and implementation consulting services. PTech Inc., can offer consulting support for all stages of the architecture development activity, ranging from the initial conceptualisation and design functions, through to the building and analysis of the architectures. The costs of these services are available from PTech Inc., and are generally decided on case-by-case bases. Currently PTech Inc. doesn't have an Australian representative, however they are seeking one, and have a commitment to the Australian region[11].

4.1.6 Who Will Use Framework?

Framework is a comprehensive tool, and it is one of the few tools in this class to have such a wide coverage. It provides support for high level organisational and business modelling, right through to support for generating Java or C++ code to implement for the various classes modelled. Framework can be extended to provide support for new

---

[10] Telephone Conversation with Larry John, Government Account Manager, PTech Inc. 22nd of April, 2001.

[11] Telephone Conversation with Larry John, Government Account Manager, PTech Inc. 22nd of April, 2001.

modelling approaches, and is one of the few tools to provide support for C4ISR Architecture Framework.

Given Framework's comprehensive coverage, as well as its ability to be customised and extended, Framework would be useful to capability developers, operational planners, and architecture researchers. For capability developers, Framework would provide the ability to capture and manipulate architectures. For operational planners, Frameworks query and template language would provide a means of extracting and analysing data. For architecture researchers, Frameworks ability to be extended and customised would provide them with a tool to support the exploration of architectures.

However, Framework's biggest failing for these three user communities is its lack of support for concurrent multi-users. While it is possible for Framework to support multiple modellers working together, through its model import and export facilities, this adds an additional administrative overhead to the modelling process. Framework's extreme flexibility, while useful for some user groups, may be too flexible for other user groups, resulting in confused and badly structured models.

4.1.7 Overall Impression of Framework

Framework is a comprehensive application, capable of supporting a wide range of different architecture communities. Framework provides support for many of the applicable enterprise and business modelling approaches, as well as modelling approaches to support application development. Framework is also one of the few tools to provide support for C4ISR Architecture Framework. Framework is also customisable, so it can be modified and extended to support new methodologies and modelling approaches. Its query and reporting features provide the user with tools for analysing the models.

Framework's modelling development interface is not at the standard expected of tools in this domain, and the poor integration between the graphical representation of models and their knowledge base representation means Framework isn't able to provide any kind of sophisticated support to the model drawing activity, nor to automatically generate diagrams from data or queries performed on the data.

4.2 Computas AS's METIS

Computas AS's METIS is described as *a powerful visual modelling tool that helps you use complex enterprise knowledge to answer critical questions and solve business problems*[12]. METIS is methodology neutral, and is flexible enough to support any modelling or architecture approach. METIS's key focus is on representing and visualizing the architecture, and as a result, METIS has a powerful model development interface. METIS can also be integrated with third party tools through its import/export wizard.

---

[12] METIS Help File Version 3.0.2.

As well as the METIS modelling engine, Computas AS also ship a WWW browser add-in, the METIS Model Browser, which allows the models developed in METIS to be accessed via the WWW.

The version of METIS described in this review was version is 3.0.2. The next version of METIS, version 3.1, is currently in Beta testing, and is targeted for release at the end of March 2001.

METIS is only available for Microsoft Windows (Win95, Win98, WinNT, and Win2000). METIS is shipped on a CD, and installing METIS is a simple and straightforward process. One of the unique features of METIS is that it is based entirely on XML (eXtended Markup Language). XML is used to represent the metamodels (called templates in METIS) as well as the user-developed models. Rather than having a single repository file, METIS includes several directories of XML files containing the descriptions for the various objects, relationships and icons (symbols) used by the default METIS metamodels.

Almost all the METIS documentation is in electronic format, apart from a lightweight *Getting Stated Guide*. Computas AS provides most of the METIS documentation in Windows Help File Format, and several PDF format documents. Overall, the documentation is well written and clear. However, Computas AS don't provide any documentation on the more advanced features of METIS, for example developing metamodels. Documentation on these features is only available via a Computas AS training program. As well as the electronic documentation, METIS is also shipped with an excellent Computer Based Training (CBT) tutorial. The tutorial includes about 30 minutes of 'screen cams' with high quality narrations, as well as summary and review pages. However, the CBT tutorial only covers the basic features of METIS.

4.2.1 Supported Methodologies and Models

The only modelling approach shipped with METIS is the GEM (Generic Enterprise Model) framework. GEM is an interesting, Zachman-like framework for modelling many different elements of an enterprise. Computas AS, have several other modelling approaches available, including IT management, systems engineering, project management and product management. These additional modelling approaches are available from Computas AS at an additional cost.

The standard GEM framework is flexible enough to capture the essence of many other more specific modelling approaches. GEM is organised into 15 different domains, with each domain containing a collection of objects, relationships and criteria (or search filters) aimed at modelling part of the enterprise. The objects contained within the different domains can be related to each other, resulting in a detailed, multi-faceted view of an enterprise. The details of the framework are discussed in more detail in Appendix C.

4.2.2 Model Developers Interface

METIS's Model Development Interface is the tool's most powerful asset. The interface is intelligently designed, mature and includes many useful and unique features. The layout, as shown by the screen shot below, includes a navigation area, called the tool tree by METIS, on the left side of the screen, and a model diagramming area, called the editor space by METIS, on the right. METIS makes extensive use of tabs along the bottom of the tree tool, and the editor space. The tabs are used to switch between different views of the tree tool and different views within the editor space. Along the top of the METIS screen are the various contextual tool bars and drop-down menus.

Overall, the METIS model development interface follows most of Microsoft's Windows 95, 2000 and NT 4.0 GUI conventions.
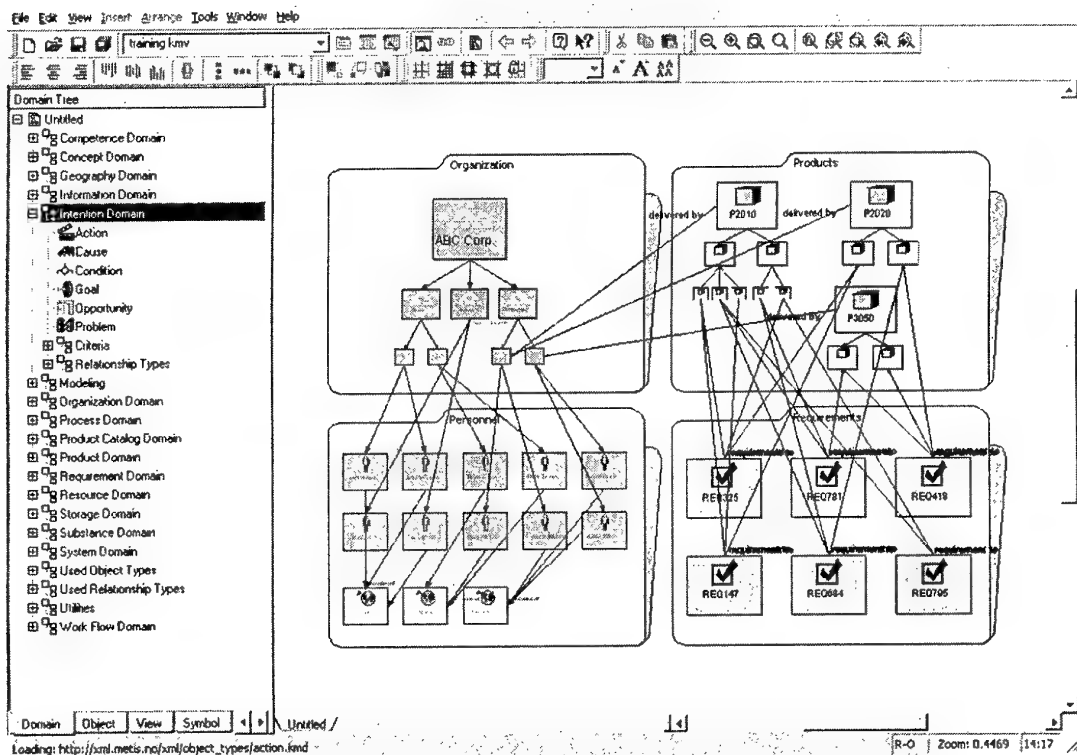
Figure 2. Computas AS's METIS, showing a GEM based model.

The tree tool (the left of the METIS screen) functions as the key navigation tool, and as the tool palette, and icon storage area. The tree tool can be switched between six different views, each associated with a specific tab running across the bottom of the tree tool. The File Tab (not shown in Figure 2) displays the computer's file hierarchy, and is a convenient way to find the various data files created by METIS. The Domain Tab is essentially the tool palette. It contains the various objects, relationships, methods and criteria made available by the metamodel being used. The Object Tab holds a list of

all the objects available in the currently loaded, and previously loaded models. The View Tab displays all the objects in the current model view. The Symbol Tab holds the various icons and graphics currently loaded. Within this view of the tree tool, existing icons and graphics can be modified and new icons and graphics created. The Loaded Tab (not shown in Figure 2) displays and provides access to all elements have been loaded by METIS. This will include the objects, relationships, icons, and so on.

The METIS modelling interface is built on the concept of a view. A view is the graphical representation of the model. The view may be the complete model, or it may be only part of the model. One model can be spread across several views. The different views of the model generally can be navigated by the tabs running along the bottom of the editor space, shown on the right of the METIS window. As will be described later in this review, views can be automatically generated as a result of a search function, by importing data into METIS.

Building models in METIS is straightforward. The various modelling objects are dragged from the domain view of the tree tool, and dropped into the editor space. Objects can be dragged and dropped onto each other to activate the default relationships. Other relationships between objects can be dragged from the tree tool to the various combinations of objects in the editor space. METIS enforces the rules of the metamodel, only allowing legal relationships between objects.

Unlike many tools in this class, the icons used to represent the modelling objects are not just simple static icons, but complex interactive elements. Icons within METIS can function as containers or as the parent of hierarchically decomposed objects. Conceptually, container objects contain other objects. For example, the four yellow folders shown above in Figure 2 are container objects that contain four different detailed model representations. Container objects can be visually opened and closed, hiding or exposing the models they contain. The type of container objects supported by METIS depends on the modelling approach being supported. For example, when developing a process model using the GEM metamodel, a process is a container object. Sub-processes can be contained within the top-level process object. The top-level process object can be opened and closed, hiding or showing its sub processes. This is a very powerful, visual way of dealing with the complexity of large models. As well as the concept of a container, the METIS modelling interface also includes the concept of a hierarchy. As with the concept of a container, a hierarchy can be used to hide or show its child elements. For example, the organisational decomposition, contained in the top left container, in Figure 2, is a hierarchy object. Closing a hierarchy object, will remove its child elements from the view. Opening a hierarchy element will show its child elements.

In addition to their graphical representation, each object within METIS also has a textual view. The textual view generally holds various additional properties of the object, for example, its name, description, and any other text fields relevant to the object. The textual view of the object also includes a dynamic list of the various

relationships in which the object participates. These links can be navigated, and at the end of the links display the text view of the object.

One of the more interesting interface features of METIS Model Development Interface is its automatic diagram layout function. The automatic diagram layout function will automatically redraw a model diagram, optimising its layout based on the layout strategy used. METIS supports two types of layout strategies; matrix and hierarchical, each optimised for particular model layouts. Different strategies can be applied to different parts of the same model. Most of the properties of the existing layout strategies can be modified, so the layout strategy can be customised for particular model types, by the end-user.

The icons (called symbols in METIS) used to represent the modelling objects and relationships within METIS can be fully customised. METIS's icons are Scalable Vector Graphics (SVG) format, and METIS includes its own icon editor to create and modify icons. Most popular graphical formats including Bit Map (BMP) and JPEG (JPG) and Windows Meta-Files (WMF), can be imported into METIS icon editor, and used as METIS icons.

Another powerful, and unique feature of METIS, is the ability to have icons, representing various modelling objects and relationships, change depending on the state, or the value of properties, of the objects they represent. Icons representing complex structures, for example containers, or hierarchies, can be visually opened and closed, with different icons used to represent the open and closed state of the object. The individual elements that make up icons (the text, graphics, colours, and so on) can change their state depending on the values of the underlying object's properties. For example, it is possible for the background of an icon to turn red to show incomplete data, and to turn green when the data is complete.

The METIS model development interface also includes the ability to zoom in and out of the model. The ability to zoom in and out of a model is a useful, visual way of navigating a large and complex model. Within METIS the model can be globally zoomed, zoomed to a selection -- so that a selected part of the model is optimally sized, zoomed to text -- so that the model is re-sized so the text is readable, and zoomed to the primary object - which re-sizes the model so a key object, a container or parent of a hierarchy is visible.

In addition to zooming METIS also includes *browse* and *fly through* modes for viewing a model. The browse mode is a powerful and sophisticated way of navigating through a model. Within browse mode, clicking-on any model object will bring it into view. Zooming out of that object will return to the previous view. The mouse can be used to move the model within the editor space. The browse mode is ideal for navigating or demonstrating large and complex models. Similar to browse mode is fly through mode. Fly through mode allows the user to easily pan and zoom in and out of the model, allowing quick, seamless navigation within a model.

4.2.3 Automation, Customisation and Analysis

METIS provides some automation, especially in the area of automatic model generation. Models can be generated from imported data (described later), or via searches on existing model data. As discussed previously, METIS uses the concept of a model view, which is a graphical representation of the model. In METIS, new model views can be generated as a result of a search function. METIS includes standard search functions, as well as search functions tied to specific metamodels. New search functions can be added by the user; very complex search functions can be built from a simple set of primitives using a simple visual interface. When the search function is executed, METIS will build a new view based on the results of the search function. Depending on the search function executed, the new model view may be a simple collection of object, or a complete sub-section of the model.

The *Action Button* object within METIS can be used to provide some limited form of automation. The Action button is a graphical device which can be placed on a model, and assigned to any one of a number of different activities; for example opening a document, or zooming to a particular object, executing a search function, or executing a METIS menu command, and so on.

METIS can be integrated with other applications through its import/export functions. METIS is able to import and export data in Comma Separated Value (CSV) format. The importing and exporting of data is governed by a CSV rules. When importing data into METIS, the CSV rules describe the relationship between the different data elements, the metamodel objects they map to, and so on. When exporting data the CSV rules describe what data will be exported, and how it will be ordered in the CSV file. Computas AS provided the CSV Rule Wizard to help with the development of CSV rules.

As well as supporting the modelling approaches provided by Computas AS, METIS also provides support for the development of metamodels (called templates within METIS) which allows METIS to be customised to potentially support any modelling or architecture approach. Within METIS, metamodels include not only descriptions of the objects and the relationships between them, but also descriptions of the different icons (called symbols within METIS) used to represent the objects and relationships, and the search criteria relevant to the objects within the metamodels. (Search criteria, and how they are used within METIS will be described later in this review). The metamodel development function of METIS was not examined in this review, because the documentation and tools needed to perform metamodel development were not made available by Computas AS.

4.2.4 Repository and Deployment Architecture

As discussed previously, METIS is based on XML, both the models developed by the user and the metamodels (or templates) are represented in XML. As a result, METIS doesn't have a single repository, but stores all the different objects, relationships, and symbols used in the metamodels, as a very large collection of individual files. The models generated by the user are stored in a single file (one for each model), with the symbols, search functions, and the like, stored as separate files. METIS doesn't support any repository functions, such as versioning/change control, access control and so on.

METIS is also a single user tool. It provides no support for multiple users, and unlike some other single user tools in this class, provides no direct way of supporting modelling in team environment.

While interactive collaboration is not possible within METIS, it is easy to distribute read-only versions of the models developed in METIS over the WWW. The METIS Model Browser is a Web Browser plug-in, and provides a powerful method for viewing and navigating METIS models. The WWW version is able to keep many of the same navigation devices as the standard version of METIS, and allows the models to be viewed and navigated graphically or via their textual interfaces. The structure and organisation used by the METIS Model Browser is very similar to that used by standard METIS. However, the METIS Model Browser is read only, the models cannot be changed or modified via the METIS Model Browser.

4.2.5 Cost and Vendor Support

Compared to many tools in this class, METIS (including the GEM metamodel) is in the medium price range, at $US 4,900 per licence. Volume discounts (ranging from 5% to 30%) are available. Annual maintenance per licence starts at $US882. Additional metamodel sets are available at $US2,500 each, with volume discounts available. Annual maintenance per metamodel set starts at $US400 per metamodel licence. The METIS Model Browser starts at $US25 per user, with volume discounts of up to 30% available. All training courses cost $US5,000 plus instructor travel and expenses, courses are limited to 10 students. Computas AS provides the majority of its support via telephone and e-mail, as well as site visits by Computas AS representatives. Computas AS doesn't have an Australian representative, and the company is split between the US and Norway, with its development centre based in Norway, and its sales and marketing office is based the US.

4.2.6 Who Will Use METIS?

METIS is a flexible, sophisticated and mature tool, which would be well suited to all three user communities. Capability developers and operational planners would find METIS an idea tool for capturing, representing and manipulating complex defence architecture. METIS's Extendability would mean it is able to support any standard or

customised architecture approach. METIS' ability to generate new models based on the powerful search functions, would allow the two groups to easily manipulate the architecture they develop. However, METIS' lack of sophisticated architecture analysis and reporting functions may limit METIS' utility for these groups, making it difficult to extract data from METIS, or to perform any kind of analysis on the data directly in METIS. For Architecture researchers, the METIS expressive metamodelling language, and its complete flexibility would provide an ideal tool for exploring architecture concepts and constructs.

However, like many tools in this class, METIS current single-user focus is a major drawback for all three groups. It would be difficult to use METIS in a fully collaborative setting.

4.2.7 Overall Impression of METIS

Overall, METIS is a sophisticated, stable and mature product. The model development interface is the best seen in this class of tool, providing some powerful and unique visual tools to capture complex architectures, including the ability to open and close objects, as well as the useful zoom, browse and fly through methods of viewing architectures. METIS's metamodelling approach seems to provide full flexibility.

However, METIS lacks of any kind of sophisticated architecture analysis tools. While it is possible to export data out of METIS into third-party analysis tools, this adds an extra level of complexity to analysing architectures. METIS' lack of any kind of central repository would make it extremely difficult to use METIS in a fully collaborative way, also the lack of central repository, and reliance on XML as the method for representing models and metamodels may impact on how well METIS can scale for very large and highly complex defence architectures.

4.3 Popkin Software's System Architect 2001

Popkin Software describes System Architect as a *comprehensive and powerful modelling solution designed to provide all of the tools necessary for development of successful enterprise systems*[13]. System Architect is a single tool that supports a variety of modelling approaches, the majority of which were designed to support the development of software. System Architect exists as a single user, and as a multi-user tool. Overall, System Architect is an easy to use and comprehensive Architecture/CASE tool. However, System Architect isn't very flexible or extendable, and users have limited ability to customise the tool or methodologies.

A single user, evaluation copy of System Architect was obtained on CD from Popkin Software. System Architect was easy to install. The evaluation copy didn't come with any hard-copy documentation, but did include a detailed, online, tour of the product.

---

[13] Popkin Software WWW Site: http://www.popkin.com.

The tour demonstrated how to create business, logical, physical and UML models. The online documentation covered all of the basic functions of System Architect, as well as describing some of the more advanced features such as reverse code engineering and reporting facilities. Overall, the online help was very useful and well written. However, some of the screen shots included in the help file didn't match the actual product, and some of the commands and options described in the documentation were not accessible in the evaluation version.

4.3.1 Supported Methodologies and Models

System Architect supports a number of modelling approaches, all of which were accessible on the trial version of the software. There is a distinct software development focus, with support for both object orientated modelling (UML, OMT, Coad/Yourdon, Booch 94 and Shlaer/Mellor) and structured analysis and design (Gane & Sarson, Ward & Mellor, Yourdon/De Marco, SSADM IV and Information Engineering). There is also limited support for business modelling, primarily through the support of IDEF standard modelling approaches. The modelling approaches supported by System Architect can't be modified by the end-users. The review examined the business, logical, physical and UML (version 1.1) modelling approaches.

Overall, System Architect implemented the various modelling approaches well. Each of the methodologies has been augmented with object descriptions that are focused around System Architect's code generation facilities. System Architect also provides some support to allow a user to switch between different modelling approaches, for example switching between a Gane & Sarson modelling approach and a Yourdon/De Marco modelling approach. However, it seems that the modelling approaches have been limited or slightly modified to enable the conversions between the different modelling approaches. In addition, the changes in methodology won't actually take effect until System Architect has been shut down and restarted. Once restarted, the tool bars and drawing options, as well as the model will be changed to reflect the new modelling approach.

4.3.2 Model Developers Interface

System Architect's model development interface, shown below in Figure 3, is divided into four main areas, the toolbar and menus along the top of the screen, the main browse window in the top left of the screen, the browse detail in the bottom left of the screen with the main work area on the right of the screen. The toolbars and menus along the top of the screen are typical for a Microsoft Window's application. The tool bars change with the current methodology, as do the drawing options that are offered in the draw menu.
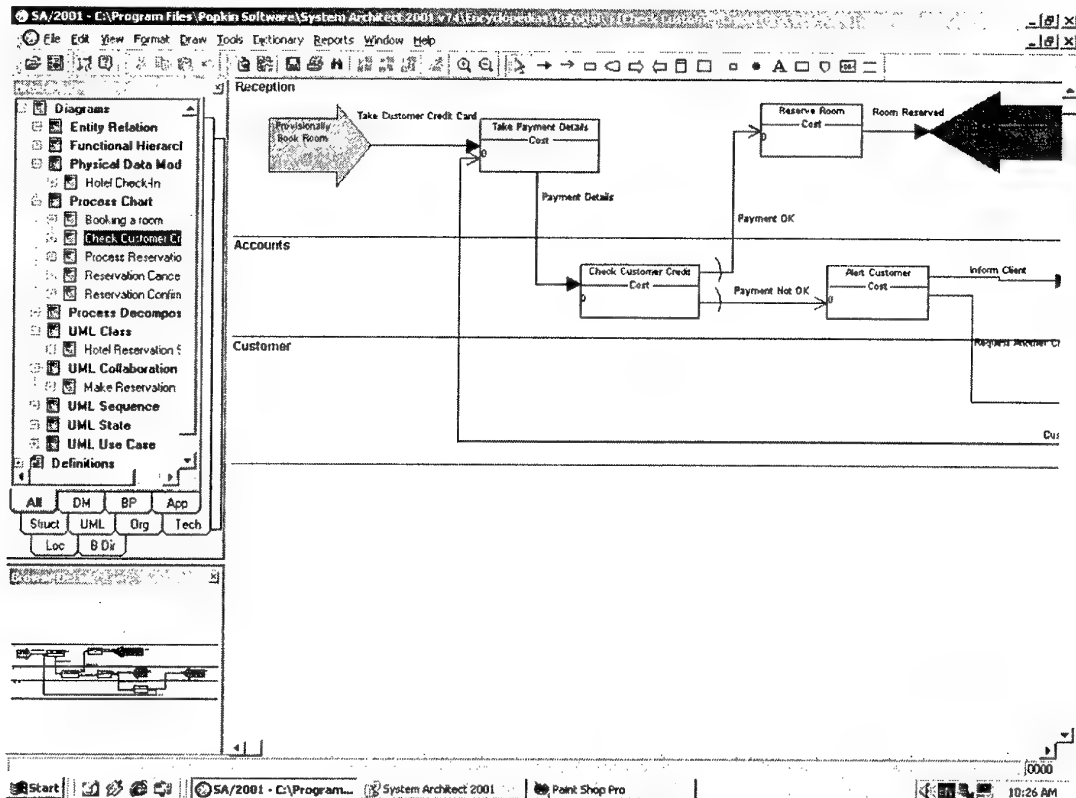
Figure 3. Popkin Software's System Architect, showing a simple process model.

The main browse window, in the top left of the screen, allows users to easily alternate between diagrams stored in the repository (called the encyclopaedia in System Architect). The browse window includes several tabs running along the bottom. These tabs hold different views of the browse window, for example, the *All* tab (opened in Figure 3) shows all the diagrams in the repository and all of the definitions relating to the currently selected methodology. The various other tabs perform similar functions.

The browse detail area, in the bottom left, holds a thumbnail view of the entire model. If the diagram is larger than the visible work area, the thumbnail view allows users to get a view of the whole diagram. This is a unique and very useful feature of System Architect.

The last area in this interface is the work area, the right side of the System Architect screen. The work area can be easily resized and scrolled. Unlike many of the other tools evaluated, there are no tabs at the bottom of the work area to allow switching between different opened diagrams.

Creating models in System Architect is similar to most other tools in this class. Objects are selected from icons in the draw menu, or the relevant tool bar with the mouse. The objects are located on the work area by clicking the mouse in the desired location in the work area. Multiple objects, of the same type, can be created by using the standard cut

and paste functions. Relationships between objects are added by clicking on the association type and drawing a line between the two objects. Users are also able to define an object once and then use it in multiple diagrams. The amount of information that is displayed on the model can be changed easily, for example hiding the attributes in a class diagram.

The icons used to represent the various objects and relationships can't be replaced with custom clip-art. However, superficial changes, such as the size and colour of the objects can be made.

While there isn't an automatic layout feature, it is relatively easy to move the objects around in the diagram. The user manual proposes that it is also possible to hard code the positions of the objects within these diagrams.

System Architect's interface conforms to most of the Microsoft Windows GUI standards and guidelines. The layout of the interface is easily modifiable by moving the toolbars and resizing or moving windows.

4.3.3 Automation, Customisation and Analysis

System Architect includes two key automation features, macros and reverse engineering. System Architect supports the creation of macros in Visual Basic for Applications (VBA). However, due to time constraints, it was not possible to fully review this feature. System Architect also provides the ability to forward and engineer databases from models created in System Architect, as well as reverse engineering data models from existing database definitions. This feature is more tailored towards software developers than to enterprise architects. It is also possible to export information from System Architect into a database. System Architect can also be used to reverse engineer structures from C++ and Java into appropriate models.

According to the supplied documentation, it is possible to import graphics, as Bitmap (BMP) or Windows Metafile (WMF) format, into System Architect. The graphics can then be incorporated into the models and can be assigned specific properties. Unfortunately, this feature did not work in the evaluation version tested.

System Architect's CASE tool roots mean it has a full featured and powerful code generation facility. Code for a large variety of languages, can be generated from the models developed in System Architect. Currently, System Architect supports C++, Java, Visual Basic, CORBA IDL, Smalltalk, Delphi Object Pascal, Power Builder Power Script, Java Script, HTML and VB Round trip. It is also possible to customise the code generation feature, using VB Scripts, for any other languages, or file format. As well as code generation, System Architect also supports reverse engineering of C++ and Java files. Once reversed engineered into models, the code and the models can be kept in sync, so a change in one, is reflected in the other.

System Architect's reporting and querying facilities are potentially very useful to software developers, but likely to be less useful to enterprise architects. However, the reporting and querying facilities weren't included in the evaluation version reviewed here. The help file outlines some of the reports that can be made in System Architect. They include reports that outline class associations, a report that outlines which objects are connected to specific classes and a complete listing of the objects and associations and what diagrams they are represented in. The reporting facility can also be used to check for errors within diagrams such as orphans and unconnected objects. A user of an earlier version of System Architect stated that whilst System Architect boasts an extensive reporting facility it would often fail to print all the information in the reports. For example, not all of a class's attributes would be included within a report. System Architect also allows users to test and modify a collaboration diagram by using queries generated within System Architect. It is also possible to query information that is stored in the database; the results of these queries can be reported in a Word format.

4.3.4 Repository and Deployment Architecture

System Architect exists as both a single user/stand-alone product, as well as a multi-user/network enabled product. When operating System Architect over a network each user has a unique repository (called an encyclopaedia), as well as having access to the group repository. The user's unique repository holds their local diagrams, and local changes, while the group repository holds the definitive, shared version of the models. Users are able to make changes to local diagrams in their repository. These changes can then be posted to the group repository, and used to update the definitive shared version of the model. Only one user is able to update models in the group repository at a one time, but multiple users can access the group repository while an update is occurring.

System Architect has several security and version control features that can be utilised when running over a network. The security mechanisms include a locking feature that can be used to bar access to certain repositories or to prevent users from modifying these repositories. The version control mechanisms that are included give users the ability to roll back and roll forward between different versions.

System Architect makes use of third party databases as its repository. Currently, System Architect is able to support the following databases: AS400, DB2, dBase, INFORMIX, Ingress, InterBase, Microsoft Access, ORACLE, OS/2, Paradox, Progress, RDB, SQL Anywhere, SQL Sever, SQL Base, SYBASE Adaptive Server Anywhere, Sybase Adaptive Sever Enterprise, Teradata, WATCOM, and XDB.

4.3.5 Cost and Vendor Support

System Architect is a reasonably priced tool. Single licences are $US 3,747.50, with the network version of System Architect able to support a floating licence policy. SASimulation, a useful add-on to System Architect, that lets its users run simulations

on any changes that they may wish to make to their model, costs \$US 818 per license. Annual updates and support for System Architect cost \$US 637.50 per licence per annum and the SASimulation Annual update/support is \$US 145.50 per annum. These update and support agreement entitle subscribers to phone support, and free updates and bug fixes. Australian based training opportunities for System Architect 2001 are currently being developed.

4.3.6 Who Will Use System Architect?

System Architect is a sophisticated and powerful CASE tool, which has continued to develop support for enterprise modelling. Its current incarnation, System Architect 2001, has continued that trend, by supporting process and information modelling.

However, System Architect is still not a flexible tool and consequently would not be suitable for architecture researchers or operational planners. While process and information modelling is supported by System Architect, defence modelling approaches, such as C4ISR architecture framework are not. Overall, System Architect is only suitable for application developers, although some capability developers, particularly those involved in developing and acquiring software systems, may find it useful.

4.3.7 Overall impressions of System Architect

System Architect is a sophisticated modelling tool. It has an intelligent, logical model development interface that can be tailored by an individual model developer. It also supports a wide variety of modelling approaches, the majority of which are related to software development. System Architect supports code generations and reverse and forward engineering to and from databases. The add-on SASimulation provides Systems Architect with the ability to simulate any desired changes to models.

However, System Architect is an inflexible tool, that can't be extended by end-users to provide support for new modelling approaches. While System Architect is a useful tool for the development of software, its enterprise focus is still too limited for any defence architecture development effort.

4.4 Proforma Corp.'s ProVision

ProVision is designed to *help companies visualize, understand and improve their business processes*[14]. ProVision consists of a suite of tools to model, analyse, and communicate the business processes, supporting processes and supporting technological systems. The tool suite supports a variety of modelling paradigms, including user-defined approaches through variations to a standard set of model types. Modelling rules are

---

[14] Proforma WWW Site: http://www.proformacorp.com.

strictly enforced, ensuring consistency, but limiting flexibility. ProVision's interface is easy to use, but has few options for customisation.

The ProVision suite of tools may be thought of as a main product with a series of add-ons. There are two variants to the main product – BusinessPro and EnterprisePro. BusinessPro provides support for defining business objectives, processes and organisational structure. EnterprisePro includes all the functions of BusinessPro, as well as adding support for modelling software applications.

Proforma Corp provide a number of add-on packages for both versions of ProVision and these add-ons include:

- AnalyserPro -- an analysis tool;
- BOSS -- a multi-user repository;
- WebVision -- which provides read-only web access to live ProVision repositories;
- JDE Exchange -- which provides J D Edwards *Solution Modeller* in ProVision and allows information to be exchanged with their One World software;
- Data Exchange -- which allows information to be exchanged with Visio, Rational Rose, ERWin, MS Project, C++ and DLL files; and
- Pro Guide -- a set of *best practice* models.

The more comprehensive EnterprisePro was evaluated in this review. EnterprisePro is a stand-alone, single user product. However, with the addition of the BOSS repository management add-on, EnterprisePro can become a network based multi-user tool. Installation of EnterprisePro was straightforward after the software had been downloaded from the ProVision website. The computer needed to be restarted once to complete the installation. In contrast, installation directly from Proforma Corp's website was not successful, as there were errors finding the correct files.

The online documentation provided with the evaluation copy of EnterprisePro was generally quite good. It included two Guided Tours – a longer version, and a shorter version called QuickStart – a Glossary, a Help File, and a Tip of the Day feature. The longer Guided Tour was a useful introduction to EnterprisePro and was generally easy to follow. However, there were some discrepancies between the tour and the product, and some concepts and icons were not introduced before they were used. For example, the Guided Tour asked you to select a specific organisation, without saying what an organization looked like. In most cases, this did not cause any difficulty either because there were limited options or because the icons conformed to MS Windows standards. However, we did have some difficulties with the auto-layout feature because of discrepancies between the documented and actual behaviour of the tool.

4.4.1 Supported Methodologies and Models

ProVision provides support for generating models using several modelling approaches including; Booch, Coad, Core, Information Engineering, J D Edwards, Jacobson Use

Case, Martin/Odell, Object Thinking, OMT, Rummler/Brache, Shlaer/Mellor and UML.

Unlike most other tools in this class, all models in ProVision are variants of one of 15 standard model types. The different modelling approaches supported by ProVision are implemented as variations of one of the standard modelling types. By default, all models developed in ProVision are developed in one of the standard 15 modelling types. The enforced mapping to the 15 standard modelling types imposes a rigid set of modelling constructs and rules, and also constrains the types of models that can be developed in ProVision. For example, several entities, such as Locations, can only be organised hierarchically. This makes it impossible to express some relationships; such as an island is part of both NSW and the Great Barrier Reef. Trying to add a new *parent* relationship removes the existing relationship. This enforcement of conceptual integrity of the models is very useful when ProVision's modelling constructs match yours, but otherwise limits the applicability of ProVision.

Because all the modelling approaches supported by ProVision are simply variants of the 15 standard modelling types, it is simple to switch between different modelling approaches by simply selecting the desired modelling approach. Once the desired approach is selected, the diagram and the toolbars will change to reflect the new modelling approach. This approach makes it easy to change the modelling approach used to represent a model, but it also means that some of the subtle variations between methodologies, and features common in other modelling tools, may be omitted. For example, there is only limited control over the visibility of attributes and operations in UML class diagrams. The visibility of the attributes and operations is determined at the object, rather than the diagram level. Furthermore, there is no concept of private attributes and operations that can be hidden while public attributes and operations remain visible.

4.4.2 Model Developers Interface

The Development Interface, shown in Figure 4, is similar to most other tools in this class. The Development Interface consisted of a navigation area (to the left), and drawing area (to the right), as well as a set of toolbars for creating, formatting, and manipulating the diagrams running along the tops and sides of the window. The navigation area consists of two areas -- a view of the projects in the repository, and a view of the models for the current project. The organisation of the models in a project was a little confusing. Five views were available – a project view, a model view, a nested model view, an object view and a scenario view. This made it easy to find the diagrams once you were familiar with the five areas. Tabs were very useful in the drawing area, where they are used to quickly switch between different models.

One unusual feature was that default positions of toolbars specific to a diagram type often appeared to the right of, or the bottom of, the drawing window (as in shown Figure 4). This tends to retain a reasonable sized drawing area, but it is confusing and

difficult to find the toolbars until you are familiar with the product. However, it is simple to move the toolbars if desired.
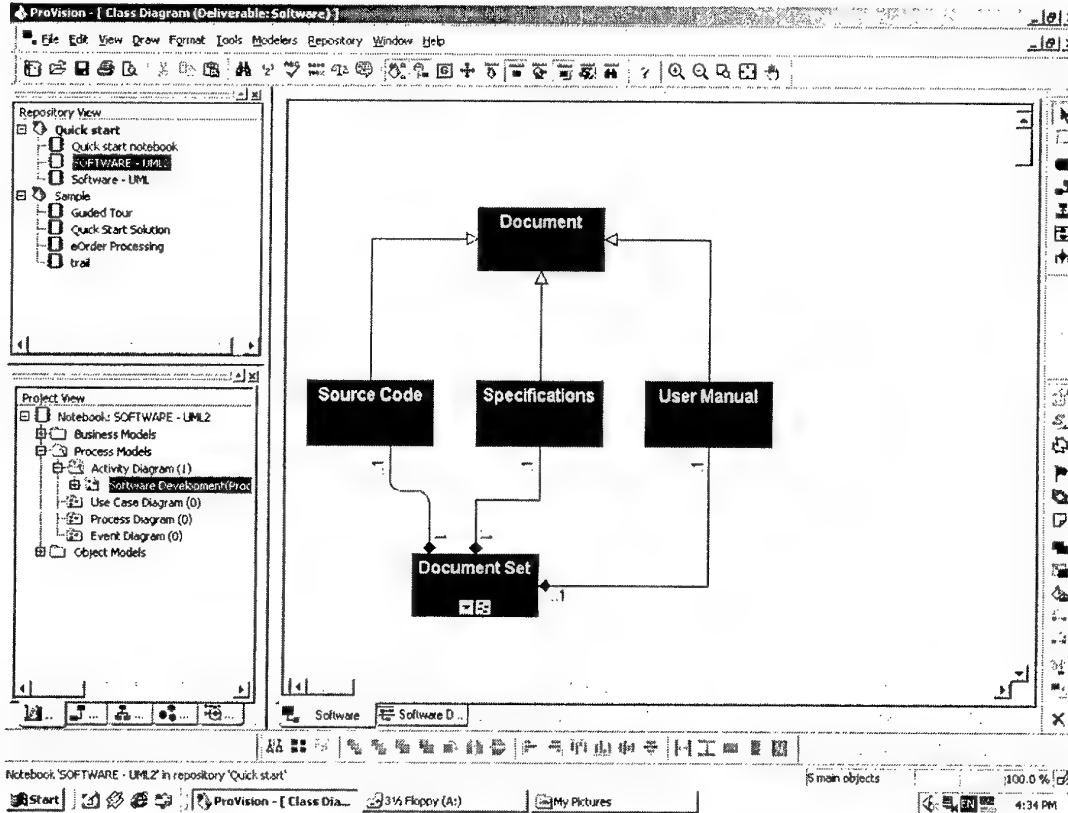
Figure 4. Proforma's ProVision, showing a simple UML diagram.

In general, ProVision was easy to use and followed the Microsoft Windows GUI standards. The standard toolbars had a similar look and feel to those used in Microsoft Windows products, and common features such as cut and paste, used standard icons and shortcuts.

Models are developed using a point-and-click approach. Objects or relationships are simply selected from the tool bar and positioned on the drawing area. When adding objects, it is possible to refer to existing entities or to create multiple new entities. Simple textual information, for example the object's name or description, can be also associated with each object and relationship.

New models are created by opening the appropriate modeller, or by extending existing models. ProVision has a good approach to model development. Unlike many other tools in this class, which are driven by specific diagrams, ProVision allows custom views of a model to be quickly created. Relationships created in one model can be viewed in another, related model. Once created, relationships can be hidden with a click of a button, or viewed by clicking on one of the grey buttons associated with an

icon (see *Document Set* in Figure 4). When viewed this way, the relationships will either be added to the existing diagram, or a new diagram will be opened if the relationships are associated with a different type of model. The display of relationships is independent of where the relationships were created.

Some of ProVision's initial displays are poor aesthetically. However, it is easy to rearrange the diagrams manually, or using the auto-layout facilities. The auto-layout facilities can rearrange the layout of the entities and relationships, or just the layout of the relationships. However, it is impossible to fix some relationships while rearranging others.

Generally, the creation and manipulation of diagrams in ProVision is relatively easy and straightforward. However, there are some inconsistencies. For example, auto-resize will only increase the size of an icon, and will continue to increase the size of an oversized icon; *undo* only applies to certain operations; and the menu items each appear five times in text formatting window. Another annoyance is that sometimes the behaviour of a tool is context dependent – in an unexpected way. For example, sometimes auto-redraw will only do a complete redraw.

One limitation of EnterprisePro is that the primary interface with the knowledge base is visual, and linked to the standard model types. Relationships are only visible in certain types of models, and often the display of relationships is only linked to certain objects. For example, workflow diagrams are associated with processes and it is impossible to quickly determine which organisations conduct an activity, and what activities are conducted by an organisation.

4.4.3 Automation, Customisation and Analysis

The format of entities, both textually and graphically is easy to change either for an individual entity, or for a modelling approach. Using your own graphics may be problematic, as the review team was unable to successfully import graphics into ProVision. It is also possible to check the spelling of entity names etc. However, while multiple dictionaries are supported, multiple languages are not.

ProVision also provides some simple facilities for checking your models. Standard checks include checks for: orphans, missing descriptions, unused objects, hidden objects, component objects, missing properties, missing custom properties, and incomplete links. EnterprisePro also allows two projects to be compared. Additional analysis facilities are available with AnalyserPro. However, due to time constraints AnalyserPro was not reviewed.

ProVision provides some facilities for exporting and importing data. Data can be exchanged (statically) between EnterprisePro and MS Access. Furthermore, reports can be written in Access and run from within EnterprisePro. This may make it possible to perform additional analysis in MS Access. Additional facilities for importing and

exporting data require the Proforma Corp's add-ons, JDE Exchange and/or Data Exchange.

4.4.4 Repository and Deployment Architecture

Both the BusinessPro and EnterprisePro versions of ProVision are single user, stand-alone, products. Both versions make use of a proprietary, single user repository, which is tightly coupled to the application.

Proforma Corp. offer an extension to ProVision, called the Business Objects Server Solution (BOSS). The BOSS adds concurrent, multi-user features to ProVision, and allows ProVision models to be shared among team members. The BOSS also supports a Check-in/check-out locking protocol, which enforces concurrency protection at the object level. The BOSS also includes basic account administration, and configurable user permissions.

4.4.5 Cost and Vendor Support

ProVision is an expensive tool suite. While the basic versions BusinessPro and EnterprisePro are relatively cheap, retailing for $US 1,995 and $US 2,995 respectively, they only provide limited functionality. The multi-user repository retails for $US 25,0000, and AnalyserPro retails for $US 995, as do Data Exchange licences – one licence is required for each of Visio, ERWin, MS Project, Rose, and C++. WebVision prices depend on the number of users, with the minimum cost being $US 20,000 per annum, for up to 50 users per day. Bulk purchase discounts are available.

ProVision has no distributors in Australia, but have existing users in Australia and New Zealand. Support is provided via email to their head office in Detroit, and through an on-line database of problems and solutions. Upgrades are available on-line. Training and mentoring are available for $US 2,500 and $US 1,500 per day respectively; expenses are extra. Training is available for groups of up to 12 students onsite or in the USA.

4.4.6 Who Will Use ProVision?

ProVision is suitable for users that want to implement standard modelling approaches. ProVision and the US DoD are investigating changes to support the C4ISR AF. They believe that little change is required. Since the C4ISR AF was designed to support C4ISR development, ProVision might be suitable for Capability Developers. However, the C4ISR AF has only limited applicability to operational planning, so ProVision might not offer the flexibility required by Operational Planner, and Architecture Researchers would be constrained by ProVision's tight modelling regime.

All Defence users would want at least EnterprisePro and AnalysisPro, and serious use by Capability Developers and Operational Planners would require at least the multi-

user repository, BOSS, and WebVision. Additional Data Exchange licences would also be desirable, to enable existing data sources to be utilised, although data could be imported to ProVision via MS Access.

4.4.7 Overall Impressions of ProVision

ProVision is a reasonable basic suite of modelling tools. Once you get used to its navigation system and tool bars, it is easy to create impressive diagrams using the auto-layout features. ProVision supports a range of modelling paradigms, based on a standard set of model types. These standards capture a set of modelling rules that are strictly enforced. Unfortunately, there are no facilities in ProVision to change these modelling rules, so ProVision has limited applicability. In the future, ProVision may be of some use within Defence if they choose to support the C4ISR AF. This may be appealing as the basic ProVision tools are cheap. However, a series of expensive add-ons are required for ProVision to be a practical alternative for the analysis and distribution of Defence architectures.

4.5 Logicon Inc.'s JCAPS

This review of JCAPS was based only on information provided to the review team by Logicon Inc.

The Joint C4ISR Architecture Planning/Analysis System (JCAPS) is a non-extendable architecture tool developed by Logicon Inc. for the US DoD. JCAPS is intended to assist in the development, management and distribution of C4ISR AF products.

The current version of JCAPS, version 2, provides support for the development of most of the essential C4ISR AF products, including, OV-1 High Level Operational Concept Graphics, OV-2 Operational Node Connectivity Descriptions, OV-3 Operational Information Exchange Matrices, OV-4 Command Relational Charts, SV-1 Systems Interface Descriptions, SV-2 Systems Communications Descriptions, and SV-3 Systems to Systems Matrices.

JCAPS supports the development of graphical C4ISR AF products, as well as allowing additional textual information to be assigned to the different modelling elements. JCAPS includes basic drawing features, as well as a zoom and pan function[15] and follows most of the Microsoft NT GUI conventions.

While the current version of JCAPS provides no architecture analysis functions, future versions of JCAPS are intended to *provide the capability to analyse and evaluate operational and system architecture designs in terms of completeness, effectiveness, and availability.*[16] However, Logicon have provided no details as to what kinds of analysis and evaluation functions future version of JCAPS will be able to perform.

---

[15] JCAPS System User Guide (SUG) Beta Release, 07 August 1998. Logicon Inc.

[16] JCAPS Concept of Operations Document. August 1998. Logicon Inc.

Unlike most tools in this class, JCAPS is tied to a specific methodology, in this case the C4ISR Architecture Framework. It can't be extended by the end-user to provide support for alternative methodologies, or even to modify or enhance the support it provides for C4ISR AF.

The current version of JCAPS, version 2, can be integrated with existing IDEF style process modelling tools. So process models, described as OV-5 Activity Models in the C4ISR AF, can be importing into, and exporting from JCAPS.

Unlike many tools in this class, JCAPS is designed for a collaborative environment. JCAPS has adopted 3-tier client/server architecture. The first tier is the JCAPS client, which can run on Microsoft's Window NT, or Sun Microsystems's Solaris. The second tier is the JCAPS application. The JCAPS application is responsible for maintaining the object model, and enforcing the modelling rules. The JCAPS application tier can run on Microsoft's Window NT, or Sun Microsystems's Solaris. The third and final tier is the Data Storage Tier, which is responsible for managing the storage of the JCAPS data. The Data Storage Tier can run on Microsoft's Window NT, or Sun Microsystems's Solaris, and uses an Oracle database.

Currently JCAPS is only available within the US DoD. Several attempts have been made by DSTO and ADF representatives to obtain a copy of JCAPS; so far none have been successful. While no prices for JCAPS have been made available, it is likely that it would be a very expensive tool to implement and maintain.

4.5.1 Who Will Use JCAPS?

Given the long-term commitment the US DoD has made to JCAPS, it is likely that JCAPS will emerge as a major C4ISR AF tool. However, it is unlikely that JCAPS will be able to support alternative methodologies, or to provide a more flexible approach to general model development.

JCAPS would be useful to anyone who fully adopts the US C4ISR AF methodology; this may include Capability Developers and Operational Planners. However, it is likely that JCAPS would hold little interest for Architecture Researchers.

4.5.2 Overall Impressions of JCAPS

Overall, JCAPS seems to be emerging as a powerful tool to support the development and maintenance of C4ISR AF compliant architectures. Planned enhancements to JCAPS will see its utility continue to improve. JCAPS is also one of the few tools to be designed to support a collaborative environment, and its deployment architecture is likely to scale very well.

However, as JCAPS is tied completely to the C4ISR AF, it can't be considered a general purpose modelling tool. It is also likely that JCAPS will be a very expensive tool to implement and maintain.

4.6 Comparison of Reviewed Tools

As the reviews show, none of the reviewed tools performed well in all the functional areas. PTech Inc.'s Framework tool was one of the few tools to have functionality in almost all of the functional areas. However, as the above results show, Framework doesn't support any particular functional area well, and its poor model development interface and high cost, lower Framework's overall score. Framework has good analysis features, but limited automation.

Computas AS's METIS excelled in the model development interface functional area, as well as in the Extendability and Customisation area, however, METIS's XML structure, and its lack of a repository, seriously limit its ability to be used in a collaborative environment. METIS has good facilities for manipulating diagrams, but a limited analysis capability.

Functional Area	Framework	METIS	System Architect 2001	ProVision
Methodologies and Models	3/5	3/5	1/5	3/5
Model Development Interface	2/5	4/5	3/5	3/5
Tool Automation	2/5	1/5	3/5	2/5
Extendability and Customisation	4/5	4/5	1/5	1/5
Analysis and Manipulation	3/5	3/5	1/5	Unknown
Repository	3/5	1/5	4/5[1]	4/5[2]
Deployment Architecture	1/5	1/5	4/5[1]	4/5[2]
Costs and Vendor Support	1/5	2/5	3/5	1/5[3]
1. For the multi-user version of System Architect.				
2. With the inclusion of the BOSS multi-user repository.				
3. This includes the BOSS multi-user repository. See review for additional information.				

Table 1. Comparison of reviewd tools by Functional Area.

Despite the hype surrounding Popkin Software's System Architect 2001, it cannot be considered an enterprise architecture tool. Almost all of the methodologies and models it supports are still aimed at software development.

Proform Corp's ProVision is an interesting tool that has taken a very different approach to implementing modelling methodologies. However, this approach may not provide defence architecture developers with the flexibility and expressive power needed to capture complex Defence architectures.

Framework and METIS are the only two tools that would satisfy the needs of all three groups as shown by the user communities comparison table below. METIS's sophisticated model development interface is far ahead of Framework. However,

Framework is placed on a par with METIS because it supports the C4ISR AF, has better analysis features, and a better repository.

System Architect is likely to satisfy no one, except maybe capability developers involved in software intensive systems.

ProVision's lack of flexibility would limit its use for Architecture Researchers. However, Capability developers and Operational Planners may find ProVision useful, if the architectures they are developing can be easily expressed in ProVision's structures. If they can't, ProVision's lack of extendability would seriously limit utility to for these two user communities.

User Communities Dimension	Framework	METIS	System Architect 2001	ProVision
Capability Developers	3/5	3/5	2/5	2/5
Operational Planners	3/5	3/5	1/5	2/5
Architectural Researchers	4/5	4/5	1/5	1/5

Table 2. Comparison of tools for the User Communities Dimension.

5. Custom and Alternative Approaches

So far, this review has considered only complete, self-contained architecture tools; tools that provide all the needed functions in a complete application, or a collection of tightly related applications. Given the emergent nature of architectures within the Defence community, it may be impossible to find one single application that can provide all the functionality needed. An alternative that Defence may need to consider, is the development of a custom architecture tool, or the use of a collection of integrated COTS applications, for example, combining an extendable drawing tool, with a COTS repository, and analysis engine, to form an architecture tool.

This section reviews current custom application development initiatives within the ADO, and outlines some potential integration strategies.

5.1 Current Custom Application Development -- The C4 Analysis Tool

An alternative to existing architecture tools, is the development of a custom tool, specifically to satisfy the needs of ADF architecture developers. One current ADF custom development effort, the C4 Analysis Tool (CAT), may provide a kernel for a custom architecture tool.

The C4 Analysis Tool (CAT) was initially developed by the ADO, in conjunction with Codarra Advanced Systems to support *C4 capability analysis, simulation and planning activities*[17]. The prototype CAT has been available on the ADF SECRET network as a web-interface to a SQL-compliant database for some time and there are now plans to grow the tool into a robust general purpose architecture tool, with the intent of *supporting C4 Architecture development and visualisation, capability analysis, simulation, training and planning activities*[18].

The current version of the CAT has a good user interface and a simple navigation scheme. The basic information is associated with C4 components, systems, and entities. Diagrams can be drawn as required, or information can be viewed in forms. Data entry is primarily through forms, although it is intended that a future version of the CAT will allow data to be imported from comma delimited text files.

The real strengths of the current version of the CAT are in its management of the configuration of communications systems, and related information and security systems. One feature of the CAT tool is that information about both planned and current capability can be maintained, and alternative configurations of systems and entities can be maintained as separate *architectures*. The CAT stores quite detailed

---

[17] C4 Analysis Tool (CAT), Stage 3 Prototype User Manual Version 3.h. Draft, 21 Nov 2000. Codarra Advanced Systems.

[18] The Mature ADF C4 Analysis Tool (CAT), Statement of Requirements, Version 1.0, October 2000.

information about the communications infrastructure. In particular, it provides a very detailed representation of communications interfaces. Connecting two components involves connecting a pair of suitable and compatible ports, which are not already in use. The CAT is also designed to distribute the responsibility for maintaining this information.

Icons, in the form of bitmaps (BMP), can be associated with each of the entities captured by the CAT. A reasonably extensive set of Defence specific icons has already been collected.

The current version of the CAT has a very rigid modelling paradigm; components are combined to form systems, which are combined to form entities (or platforms). The interfaces at each level in the hierarchy are a subset of the unused interfaces at the lower levels in the hierarchy. This assumes a bottom-up modelling approach where components are defined first, then systems, and finally entities. These restrictions may be relaxed in future versions of the CAT.

The CAT has limited models of information flows, and tasks or activities. Information flows are modelled at a very abstract level, using 10 NATO standard information categories that include Operations, Plans, Intelligence, and Training. The current version of the CAT has no way to represent tasks, although tasks and missions will probably be associated with architectures, operational elements, and possibly with information flows, in future versions of the CAT.

The information flows that are captured in the current CAT can be viewed in a dynamic fashion. Information flows related to an entity, or between a set of entities, can be displayed in a circular layout that clearly shows all the relevant exchanges. The CAT can also produce a variety of standard reports based on the information stored in the repository – including a report on the limitations of a particular configuration or architecture. Future versions of the CAT will also allow some analysis of metrics that are associated with the information flows. There are no plans to allow users to develop their own reports or analysis functions.

The constraints imposed by the current version of the CAT, and to some degree the planned, future version of the CAT, limit the applicability of the tool. Some Capability Developers and Operational Planners might be satisfied with the limited information model, the need to build entities using a bottom-up approach, and limited analysis features. However, the lack of flexibility precludes the use of the CAT tool by Architectural Researchers, unless they have access to the source code. The CAT is more suited to managing the configuration of communications and related systems.

The plans for the CAT are still unclear. The existing version of the CAT may be extended incrementally, adding new features with each version. Alternatively, a completely new development effort may be undertaken. The cost of enhancing the

existing version of the CAT, or undertaking a new development effort is still unknown, as are the final functions of any future version of the CAT.

5.2 Integrating COTS Tools

An alternative approach to developing a custom enterprise architecture tool may be to integrate relevant COTS products. For example, it may be possible to integrate an existing, high-end, extendable graphical tool, such as Microsoft's VISIO, with an Object Orientated style repository, for example ConceptBase. Analysis and simulation of the architecture could be performed by specialist analysis tools, for example, ProSim/ProCap and AI0Win, as well as general-purpose analysis tools, for example, Microsoft Excel, or even custom analysis tools.

The key advantage of this approach to the development of an enterprise architecture tool is its potentially lower cost, and lower risk than a custom development. While many of the sophisticated COTS tools needed for this approach wouldn't be cheap, they may be cheaper than undertaking a custom development to produce tools of similar functionally. Development time may also be much shorter for the integration of COTS tools than for the development of a custom application.

While this approach may be cheaper and quicker than a custom development, the result of this approach may simply not have the functionality required by Defence architecture developers. COTS tools are seldom a perfect fit, and the ability to integrate COTS tools often depends on how well the various tools can be extended. This approach is also susceptible to changes in the interface and levels of support provided by the vendors of the individual products.

However, this approach may provide a useful middle ground between a commitment to a single vendor/single application, and a full custom development. Defence may be able integrate different COTS tools together, to experiment with the requirements of a full custom development effort.

6. Conclusions

Currently, there is no one architecture tool likely to meet all the needs of the three different user communities. Architecture tools are still at the point where they require considerable input from programmers, systems analysts and analysts with an in-depth knowledge of architectural models and methodologies, as well as the enterprise domain, to make them useful for deployment within a Defence environment.

The Enterprise Architecture field is still maturing, both at the business end as well as in the tool development arena. It is becoming apparent, however, that the development of enterprise architectures is a community-driven activity and so an essential requirement of tools in this area is deployability over a networked, distributed architecture. Software development communities have already learned this lesson and hence the mature networked capabilities of a tool like System Architect have allowed it to dominate in its field.

Because the Enterprise Architecture field is still immature, encapsulation of architectural approaches is rare within these tools. Both JCAPS and PTech Inc.'s Framework have made an assault on the C4ISR Architecture Framework. Framework also includes support for the Zachman Enterprise Architecture approach. The importance of these features is that it widens the tool's availability to end-users. If the end-user's task is merely one of populating an existing model with the business context and data, this is a far easier task than instantiating a metamodel with a particular approach then applying it to the business context.

The three user communities differ in their requirements from an architecture tool. Capability Developers require a powerful and intelligent repository to store a wide variety of models, metamodels and instantiated models. They also need a tool that can cope with linked documentation such as scoping documentation and capability proposals. The repository would need to have mature database rollback and recovery capabilities as well as data integrity and versioning controls.

The user interface would need to be well developed and user-friendly with the ability to customise presentation aspects such as icons and labelling. Encapsulation of an architecture framework within metamodels is another essential feature, with the model development interface minimising the user development aspects, but allowing for some domain customisation.

Currently tools with strengths in these areas are Logicon Inc.'s JCAPS, PTech Inc.'s Framework and Computas AS' METIS.

While Operational Planners also need, a powerful and intelligent repository with a facility to allow for connectivity to data feeds and to allow for automated diagram production, they would need a well developed and friendly user interface as well.

Perhaps the most important requirement of a tool for this group would be support for architecture and system analysis and completeness investigations. Models, which encapsulate compliance rules and allow queries on enforcement for analysis purposes would be most useful to this group.

Tools with strengths in there areas are ADO/Codarra's CAT tool and PTech Inc.'s Framework. The CAT tool has good, standard analysis and manipulation functions, but its focus is on communications architectures rather than Enterprise architectures. In contrast, Framework has relatively few useful standard analysis and manipulation functions, but its strong query and reporting facility allow custom analysis functions to be developed.

Architecture Researchers are a lot easier to please. They can cope with poorly designed visual and modelling interfaces, and have the ability to create models from a variety of architectural approaches. An important requirement for this group however, is a flexible metamodelling facility. This allows the exploration of new approaches, which can assist the enterprise in forging new architectural designs.

Other features that are important for this group are analysis and reporting facilities as well as automated data capture and display.

The tools that would be strongest in this area are PTech Inc.'s Framework and Computas AS's METIS.

Appendix A: Survey of Enterprise Architecture Tools

Product Name	Vendor and vendor URL	Tool Description	Methodology and Models
FirstStep	Interfacing Technologies Corporation (www.interfacing.com)	• BPR tool focused on process modelling. • Powerful tool providing a wide range of approaches to modelling processes. • Provides and simulation and *what if* evaluation tools for process models.	• Supports the CIMOSA framework. Thus it supports the modelling of Processes, Activities, Organisational Groups, resources and Materials. • Modelling entities can be customised, but new modelling types can't be added.
SILVERRUN Workbench	Silverrun Technologies (www.silverrun.com)	• Suite of tools designed to support ER and business process modelling, as well as UML and code generation. • Client/Server deployment architecture, with high levels of integration into third party tools.	• ER Diagrams. • UML. • Process Models.

Enterprizer Suite of applications	S3 International LLC (www.s3inter.net)	• Enterprise modelling and analysis suite of tools, aimed at strategic planning, and long-term scenario generation.
		• Suite of six closely related applications:
		  ○ *strategyEnterprizer* captures high-level strategy formulation, integration, alliances, joint ventures, mergers, acquisitions and spin-offs.
		  ○ *designEnterprizer* supports the design, assessment and optimisation of products, services, concepts and initiatives within the organisation.
		  ○ *knowledgeEnterprizer* knowledge capture and retrieval tool.
		  ○ *scenarioEnterprizer* supports 'what-if' scenario assessment of future scenarios, to support strategic and operational planning.
		  ○ *decisionEnterprizer* support the evaluation and prioritisation of multiple solution decision.
		  ○ *performanceEnterprizer* support for
			• The suite uses methodologies and modelling approaches developed by the vendor.
			• major parts of the suite are designed to be customised by the vendor to match the target organization's strategic environment, specific problems, and so on.

Product Name	Vendor and vendor URL	Tool Description	Methodology and Models
		implementing the balanced scorecard performance measurement approach. • All six products share common interfaces, repositories, etc. • The vendor can customise and extend all six tools, however, end-user customisation doesn't seem possible. • The tools are tightly tied to S3 International's consulting business, and are sold as part of a consulting process.	

Visible Advantage EA	Visible Systems Corporation (www.visible.com)	• Powerful software modelling and design tool.
		
		• Supports development of:
		   o High-level business plans and goals.
		   o Process models.
		   o Data models.
		   o Systems Design -- in the form of system specifications.
		
		• Also generates necessary documentation, and can performs quality assurance testing on entered models.
		
		• Supports the linking of the different models together, thus providing a link from business goal to system.
		
		• Provides good support multi-user environments with a client/server architecture and support of concurrent users.
			• IDEF0 Activity Models.
			• IDEF1X Data models.
			• Activity Based Costing (ABC).
			• Custom Process models.
			• Custom Business Strategy models.
			• Some level of customisation of methodologies and models.

ProVision	Proforma Corp (www.proformacorp.com)	• Comprehensive BPR and enterprise modelling tool. • Provides support for high-level organisation models as well as lower level process and workflow models. Supports linking between the two. • Provides powerful repository, and the ability to develop custom views of data and models. • Vendor considering support for C4ISR AF.	• Rummler-Brache. • LOVEM. • Booch. • OMT. • UML. • Martin/O'Dell. • Martin/IE. • Coad. • Supports some levels extensions and

Product Name	Vendor and vendor URL	Tool Description	Methodology and Models
			customisation to the existing models and methodologies. However, new models cannot be added. • Currently the vendor is reviewing C4ISR AF and may provided explicit support for C4ISR AF in future releases.
System Architect 2001	Popkin Software (www.popkin.com)	• Comprehensive and sophisticated modelling tool, designed initially for software development, but extending to support higher level organisational modelling. • Links high level organisational modelling with lower level application modelling. • Multi-user repository • Includes a process simulation tool that allows process models to be simulated. • Able to generate and reverse engineer application code. • Support for MS VBA to customise modelling functions, as the reporting language and to customise the tool's interface, as well as to automate functions.	• IDEF3 Process Flows Models, and Process charts. • Gane and Sarson. • Yourdon/DeMarco. • Ward and Mellor. • Information Engineer. • SSADM. • Data Models -- E/R Diagrams. • UML. • IDEF Family. • ABC based on process models. • Zachman framework. • Some level of customisation to existing models and methodologies. New models or methodologies can't be added.

WizdomWorks!	Wizdom Systems Inc
	(www.wizdomsystems.com)
		• BPR suite of applications, specifically, ProcessWorks! IDEF activity/process modelling, and DataWorks! low-level data modelling tool.
		• Collection of tools that perform different types of analysis:
		   o Process flows
		   o Activity Based Accounting
		   o Timeline analysis
		   o Total Quality Management
		   o Process simulation tool
		• Includes a Document Manager, which links documentation (internal, as well as external files) to process/activity models.
		• Both single and multi-user versions of the
			• IDEF0 activity/process modelling.
			• E/R Diagrams for data modelling.
			• IDEF1X for data modelling.

Product Name	Vendor and vendor URL	Tool Description	Methodology and Models
		applications are available.	
Corporate Modeller	CASEwise Systems Inc (www.casewise.com)	• Process/data modelling tool, aimed at BPR and systems implementation. • Supports automatic documentation generation. • Include a process simulation functionality. • Repository supports concurrent multi-users.	• Hierarchy Diagrams. • Matrix Diagrams. • Data Flow Diagrams. • Process Dynamics Diagrams. • Entity Relationship Diagrams. • Some limited support for adding additional, custom modelling approaches.
METIS	Computas AS (www.metis.no/products/)	• Enterprise modelling tool. The tool is designed to be completely extendable. It supports few standard models, but provides a powerful metamodelling facility, so end-users can add their own modelling approaches. • It supports a powerful view concept, which allows information displayed by a model to be controlled by the modeller. • Good auto draw and GUI facilities. • Current multi-user support is limited.	• Generic Enterprise Modelling (GEM). Custom enterprise modelling language. • Fully customisable and extendable, it is able to support any methodology or modelling approach.
MetaEdit+ Method Workbench and MetaEdit+	MetaCase Consulting (www.metacase.com)	• Method Workbench and MetaEdit+ are meta-CASE tools. With Method Workbench, developers design a CASE methodology. This methodology is then enforced on application developers used MetaEdit+. • Supports the development of custom visual modelling languages, code generation and document generation. • Supports a central, concurrent user repository.	• Booch -- UML. • Rumbaugh -- Object Modelling Technique. • Booch -- OO Design. • Coad/Yourdon -- OO Analysis and Design. • Coleman -- Fusion. • Henderson-Sellers -- Moses. • Embley -- OO systems Analysis. • Schlaer/Mellor -- OODLE. • Yordon -- Structure Analysis and Design. • Ward/Mellor -- Real-Time SA/SD. • IBM's business Systems Planning. • Porter -- Value Chains & Value Systems. • Goldkuhl -- Activity Analysis. • Customisable via the use of the GOPRR method engineering language. Any methodology or modelling method can be added. Existing approaches can be modified.

Product Name	Vendor and vendor URL	Tool Description	Methodology and Models
Framework	PTech Inc. (www.ptechinc.com)	• Described as an Enterprise Architecture tool. • Supports wide range of different business and technology/system development methodologies. • Provides support for code generation, and can be customised to support any OO style language. • Provides support for C4ISR AF via the use of vendor supplied add-in. • Very High levels of Extendability, via meta-modelling. • Single user only. Provides simple repository sharing functions, unlikely to support true concurrent users in short/medium term.	• Business Architecture/Value Chain Models. • Strategic Planning. • Process/Workflow Models. • UML. • Zachman Enterprise Framework. • C4ISR AF. • Several custom modelling approaches. • A customisation kit is also available for each version.
MooD	The Salamander Organization Ltd (ww.tsorg.com)	• MooD is a facilitation tool designed to support the Soft Systems Methodology (SSM). • SSM helps map out business object, process and procedures. • MooD provides support to capture the information generated by SSM. • It supports a multi-user repository. • It can be integrated with other Windows applications through OLE and ODBC.	• Soft Systems Methodology (SSM).

ARIS Product Suit	IDS Scheer Inc (www.ids-scheer.com)	• A BPR and process modelling focused suite of tools that support the ARIS methodology.
		• The vendor has recently re-positioned the tools to deal with *e-business processes*.
		• The suite consists of ARIS Toolset and several related products. The ARIS Toolset provides business process definition and analysis.
		• The suite consists of:
		  o *ARIS Easy Design* -- business process capturing tool.
		  o *ARIS Web Publisher* -- outputs models built in ARIS Toolset to WWW.
		  o *ARIS Simulation* -- simulation of process models recorded by ARIS Toolset.
			• ARIS Framework is a custom methodology developed by the company. The Framework captures people, data, activities, products, processes and objectives, and the relationships between them.
			• Additional tools (ARIS ABD and ARIS BSC) support Activity Based Accounting (ABC) and Balanced Score Card (BSC) methodologies.

Product Name	Vendor and vendor URL	Tool Description	Methodology and Models
		○ *ARIS ABC* -- ABC tool for costing processes recorded by ARIS Toolset. ○ *ARIS BSC* -- Balanced Score card tool. • One integrated repository is used by all the tools.	
BOTiCMAP suite of Enterprise Modelling tools	BOTiC Ltd (www.botic.demon.co.uk)	• A suite of enterprise modelling tools developed by the vendor to support their consultancy work. • The support their propriety method, the *BOTiCMAP Technique*. The key focus of the method is to capture organisational/enterprise knowledge and relationships between descriptions of business processes and technology to support it. • The tool is built on an MS Access DB, and supports limited scalability and concurrent usage.	• BOTiCMAP Technique, an organisational/enterprise knowledge capture methodology.
Cool: Product Suite	Computer Associates (nee Sterling) (www.cai.com)	• The Cool suite is aimed at the whole application development process from the initial design to the code generate. • Support to enterprise modelling is provided by the business process-modelling tool Cool:Biz. • Cool:Biz provides a concurrent multi-user, environment. • CA seems to be supporting most of the cool products, except for COOL:DBA and COOL:BusinessTeam.	• Business Process modelling via Rummler and Brache methodology.
ABLE PM	Triune Software (www.donet.com/~triune)	• A generic BPR tool that implements IDEF0 methodology. • Also comes in a *Reader* version, which provides read only access to the models. • The related tool *Web Publisher* allows diagrams created in ABLE PM to outputted to the WWW. • The tool can't be extended to support new methodologies or models. • The tool is structured for single users only.	• Business Process/Activities via the IDEF0 methodology.
JCAPS	Logicon, Inc. (www.logicon.com)	• JCAPS -- Joint C4ISR Architecture Planning/Analysis System. Developed by Logicon for the US DOD.	• Supports the development C4ISR AF products. Currently, supports provides support for: OV-1, OV-2, OV-3, OV-4, OV-5, SV-1, SV-2 and SV-3.

Product Name	Vendor and vendor URL	Tool Description	Methodology and Models
		• Currently the only tool designed specifically to support C4ISR AF. • Very flexible implementation architecture, and provides support for concurrent multi-users. • Integrates into US DoD hardware and software standards, data sources and networks.	
Oracle Designer	Oracle Corporation (www.oracle.com)	• Designer supports business modelling, data modelling and data flow modelling. • It allows the high-level business models to be mapped to application models. • Application code (in the form of C++, Visual Basic, Oracle Forms, and Oracle Reports) can be generated from the applications models. • Powerful Repository (which is also sold as a stand alone application). The repository supports concurrent multi-user, versioning and access control.	• Business process modelling in the form of Rummler and Brache. • Data modelling in the form of E/R diagrams. • Data Flow diagrams. • Business decomposition in the form of Functional Hierarchy Diagrams. • Also supports UML for application modelling.
BPWin	Computer Associates (nee Platinum) (www.cai.com)	• BPWin a business modelling tool. • It supports functional/process modelling, workflow modelling, and data flow modelling. • Includes interfaces to allow it to be integrated with other, external, software applications. • When combined with CA's ModelMart model management tool, BPWin can support concurrent multi-user.	• Functional/process modelling is provided by IDEF0. • Workflow modelling is provided by IDEF3. • Dataflow modelling is provided by DFDs. • Also provides support for ABC. • The properties of model elements can be extended. New modellings methods can't be added, and the semantics of the existing models cannot be changed.
ProcessWise WorkBench	Teamware -- A Fujitsu Company (www.teamware.com)	• A business process modelling and business analysis tool. • The tool supports the modelling of business processes. • Sophisticated analysis and simulation can be performed on the models. Both time and event-bases simulations are supported by the tool. • Supports version control of models, to track changes.	• Several different types of modelling, including: Process Hierarchy, Rummler and Brache, Use-Case, Workflow. • Modelling entities can be customised, so can that rules that govern the simulations. The semantics of the models can't be modified.

Product Name	Vendor and vendor URL	Tool Description	Methodology and Models
		• WWW export facilities. • A *Specialist Edition* version of the tool, which only supports pre-defined sets of modelling objects and rules, is also available.	
GRADE	INFOLOGISTIK GmbH (www.infologistik.com)	• GRADE is designed to model complex socio-technological systems. • Supports an "as is" and "to be" views. • Supports the analysis of the models through simulation and/or prototyping. • Supports export to HTML, general export and export in UML to Rational Rose. • The tool is available in single or multi-user versions. Supports merging, comparing of models, and simple change management.	• Custom methodology build on three key objects -- Active Objects (or producers), Passive Objects (resources, things produced by active objects) and processes (or activities that produce a result). • The custom methodology captures structure, dynamics, events, data, functions, resources, relationships, and situation parameters.

ProSim/ProCap and AI0Win	Knowledge Bases Systems Inc. (www.kbsi.com)	• Two tightly related modelling, analysis and simulation tools.
		   o ProSim/ProCap is a business process and workflow modelling and simulation tool.
		   o AI0Win is an activity-modelling tool.
		• Both packages support export to HTML and.
		• A reader for both packages is also available.
		• ProSin/ProCap can also be exported to project management software, for example MS Project.
		• ProSim/ProCap allows additional rules to be added to the models. These govern some aspects of how the models behave.
		• AI0Win can also be exported to Excel, as well as an ABC add-on tool from KBSI.
		• The products come as both single user and network versions.
			• ProSim/ProCap supports IDEF3 modelling.
			• AI0Win supports IDEF0 modelling

4Keeps	A.E. Experts, Inc. (www.adexperts.com)	• The 4Keeps suite of tools adds additional functional to Microsoft's Visio tool. • The 4Keeps suite provides a repository that holds each of the objects represented on the Visio diagram.	• Only supports modelling diagrams. • UML. • Booch. • Rumbaugh. • Shlaer-Mellor.

Product Name	Vendor and vendor URL	Tool Description	Methodology and Models
		• The suite also supports the reverse and forward engineering of database descriptions. • The suite also allows the different objects captured in the repository to be represented in a variety of different ways, and between different modelling representations. • A documentation tool is also included, which will product Word or HTML format documents based on the data held. • There is also an *inspector* tool, which will check models for completeness and consistency. • The tool is extensible; additional classes of objects can be added to the repository.	• Yourdon-Coad. • Martin. • Chen. • IDEF1X. • SSADM. • Bachman.

Appendix B: Review Template

The goal of the review framework is to provide a consistent approach to reviewing the architecture tools. The template outlines different areas to consider, and provides indicative questions you should try to answer. Remember to highlight any special features the tool has, or any function it performs extremely well, or extremely badly.

B.1. Tool Overview

The goal of this section is to provide the context for the more detailed review of the tools functions, and its fitness for purpose. The overview should include:

o   An overview of the intended goals, user groups, purpose of the tool; this may be available from the vendors WWW site, or in the introduction documentation provided by the tool.

o   An overview of the structure and organisation of the tool.
    o   Is the tool a single application? Or several related applications? Does the tool run over the network? Or on a single workstation?

o   The installation process -- any problems? Anything special?

o   Overview of the documentation provided by the vendor.
    o   How useful?
    o   Quality?

o   Overview of the help information provide within the tool.
    o   How useful?
    o   Quality?

o   Any other items of interest?

B.2. Tool Functionality

This section reviews the core functionality of the tool. For the tool being reviewed, try to describe the kinds of functions it has within each of the functional areas. Some typical questions are presented within each functional area, however, not all of these questions will be applicable to all tools. Remember to outline any special or unique functions the tool offers in each area, or any major deficiencies the tool may have.

B.2.1   Methodologies and Models

o   What methodologies and modelling approaches does the tool support?

o How well does the tool support each of the methodologies and modelling approaches?

o For tools that support several different complementary methodologies or approaches, how are the different approaches integrated?

o For tools that support competing methodologies, how are the different approaches integrated? Is it possible to move the one set of data between the different approaches? How well is this supported?

o For tools that support functional methodologies (those methodologies that mandate a sequence of activities), how intelligent is the enforcement of the structures?

o Any other issues or functions?

B.2.2 Model Development Interface

o Provide an overview of the model development interface.

o Quality of the design and layout of the model development interface?
   o How well does it use the screen space?
   o Is the interface logical and consistent to navigate?
   o What kind of conceptual integrity does the interface have?
   o How well does the interface follow the conventions of the host operating system?

o What kind of drawing support does the tool provide (this is also addressed by the tool automation group)?
   o Auto diagram layout?
   o Auto connection?
   o Contextual, data option menus?

o Customisation of iconography.
   o Can the icons used for entities, relationships, and so on, be customised?

o Any other issues/functions?

B.2.3 Tool Automation

o Provide an overview of the kinds of automation functions the tool provides.
   o Customisable automation functions -- Scripts, Macros, etc?
   o Built-in automation functions -- auto draw, auto layout of models, etc (see the previous group -- Model Development Interface for more info).
   o Data driven automation -- model generation based on data, or results of queries, etc.

- o Provide descriptions of any customisable automation functions the tool supports.
    - o What elements of the application or models can they be used to automate?
    - o How useful are they?
    - o Quality?

- o Provide descriptions of any built in automation functions the tool supports.
    - o What kinds of automation do they offer, what elements of the application or models can they be used to automate?
    - o How useful are they?
    - o Quality?

- o Provide descriptions of any data automation functions the tool supports.
    - o What kind of automation can be performed on the data? Mass updates? Auto loads? Can new models, entities or relationships be generated based on some kind of data manipulation/process function?
    - o How useful are they?
    - o Quality?

- o Any other issues/functions?

B.2.4 Extendability and Customisation

- o Provide an overview of the tool's Extendability and/or customisation concept.
    - o What does it apply to?
    - o How is it implemented?

- o Describe what kind of support the tool provides for customisation.
    - o How can the existing metamodel structure (or whatever metamodels map to in the tool being reviewed) be modified?
    - o How can support for new modelling approaches be added?
    - o What else can be or needs to be modified to support new modelling approaches or methodologies? For example Textual interface (ie Forms) repository queries or reports? Model constraints or rules?
    - o What kind of programmatic customisation is the tool capable of supporting?
        - What kind of language is used to make the modifications?
        - Scope of the function -- what can it affect?

- o Describe the support the tool has for Extendability.
    - o How can the tool be integrated with existing software packages?
        - Import/Export.
            - What kinds of file structures? What kinds of tools?
        - Exposed API?

- Support for middleware -- COM/Active X, CORBA, others?

o Any other issues or functions?

B.2.5 Analysis and Manipulation

o What types of analysis functions does the tool provide (also include any simulation functions in this group).
- o Are they analysis functions linked to specific models?
- o What analysis can the functions perform -- what kind of questions, about what types of models/methodologies, can the analysis functions answer?
- o Any interesting or unique analysis functions?

o What kinds of data/model manipulation functions does the tool support?
- o Can the visuals (what entities, and relationships shown) of the models be changed/manipulated?
  - If so, how? Data query? Fixed function?
- o Can different models be combined or deconstructed in some way?
  - If so, how? Data query? Fixed function?
- o Does the tool offer any other interesting manipulation functions?

o Any other issues or functions?

B.2.6 Repository

o Repository structure and type (this is closely related with the Deployment Architecture functional group).
- o What is the structure between the tool and its repository?
  - Tightly coupled, single user or single repository?
  - Simple client/server?
  - More than 2 tier client/server?
- o What kind of repository does the tool support?
  - Is the repository based on a 3rd party product -- for example 3rd party DBMS?
  - Is the repository proprietary product?

o What kind of multi-user concept does the repository support?
- o Is the repository capable of supporting concurrent multi-users?
- o Is the repository intended to be single user only?
- o Does the repository allow models to be exported and imported? If so, can it deconflict the various models if needed?

o What kind of management functions does the repository provide?
- o Versioning and/or change control?

- o The ability to roll back/forward between different versions?
- o Access control?
- o Other functions?

- o Any other issues or functions?

B.2.7 Deployment Architecture

- o Describe the tools deployment architecture.
  - o Single tool?
  - o Several closely related tools?
  - o Single workstation?
  - o Client/server?

- o What other infrastructure does the application need (in addition to the OS)?

- o Any other issues or functions?

B.2.8 Costs and Vendor Support

- o Describe the indicative costs for the tool.
  - o What kind of licensing deals does the vendor offer?
  - o What is the cost per single user/single license?
  - o What is the cost per Group/bulk/site licenses?
  - o Does the vendor support floating licenses?
  - o Are their any other approaches to lower the over all licensing cost of the application?

- o What after sales support does the vendor offer.
  - o Vendor provided training?
    - What is the cost of training?
    - Is there an Australian representative?
  - o Support agreements?
    - Type of support provided?
    - What is the cost of the support agreements?
    - Australian Representative?
  - o Application maintenance agreements?
    - What is offered as part of the application maintenance agreements (Upgrades, Bug fixes)?
    - What are the cost for maintenance agreements?

- o Any other issues or functions?

B.2.9   Special features or functions

> o Does the tool offer any special features or functions not covered by any of the above groups?

B.3. Suitability and Fitness for Purpose

The goal of this section of the review is to provide qualitative judgements of the tools suitability for various user groups within the ADO.

> o For each of the three groups (described in Section 2.2), discuss how well the tool would satisfy the needs of the user communities. Draw on the descriptions of the various functions provided by the tool.
>   - Highlight functions of the tool that would be useful to them.
>   - Highlight the functions the tool lacks that make it less useful to them.

B.4. Overall Summary of the Tool

The goal of this section is to summarise the review, and highlight any interesting features or shortcomings of the tool, and to present a value judgement on the overall performance of the tool.

> o Overall summary of the tools functionality.
>   - Re-state what the tool does well, and what it does poorly.
>   - Highlight any unique or interesting features of the tool.
>
> o Overall summary of the tools suitability and fitness for purpose.
>   - For whom would this tool be ideal?
>
> o Overall impression of the tool.
>   - Qualitative judgement of the tool based on your experience with it.

Appendix C: The GEM Framework (Computas AS's METIS)

Within GEM, the structure of the enterprise being modelled can be captured by the organisation domain, which captures the main elements of an organisation. These elements can be related to other domains within GEM, thus providing a way of contextualising the various dimensions of the enterprise, for example the process, products, information flows, and so on. Objects captured within the GEM domains can also be related to specific geographic locations via the use of the geography domain, which can be used to capture the physical locations of most of the GEM objects.

The functions of the enterprise can be captured by the process and workflow domains. The process domain supports IDEF's ICOM style process modelling. The workflow domain can be used to represent processes with information and physical flow, events, and process roles. Both process and workflow domains can be related to the organisation domain, as well as the domains used to capture information and physical objects.

Information within the enterprise can be captured by the information domain. The information domain implements a high-level information model, which includes the information element, information element type, and information element group. This structure is useful for taxonomic descriptions of information. The information, and information descriptions, captured by these objects can be related to most other domains, including the processes, systems, storage, requirement, and concept domains.

The end states, needs, and goals of the enterprise can be captured by the requirements domain. The requirement domain is used to represent requirements and solutions, as well as relating requirements to goals, functions, competencies and information.

The enterprise's physical assets, function, and competencies can be captured by the resource, storage, competence, substance and system domain. The resource domain is used to model resources, which are defined as people, machines, and tools. Resources can also be related to other domains through location, competence, systems, and positions. The storage domain can be used to model the *storage* of objects; including documents, processes, systems, information flows, and so on. The competence domain can be used to capture current skills or required skills for a resource. Competence can also be related to processes. The substance domain is used to represent physical entities; physical entities can be related to the storage domain, the process domain and in the form of an information flow, to the information domain. The final domain, the system domain, is used to model the relationships between systems, information flow, functionality, requirements and so on. Systems can be related to most other domains through various relationships.

The various products of the enterprise can be captured within the product domain, and refined by the product catalogue domain. The product domain can be used for simple product modelling. A product can be modelled as a hierarchical decomposition of sub-structures, or as a collection of modules. The product catalogue domain can be used to show the relationship between products, and the standard parts that may be used to create them.

The final two domains within GEM, the concept and intention domains are used to capture conceptual ideas. The concept domain provides a method for context independent modelling of concepts. It allows any level decomposition of a concept, much like a mind-map. The intention domain is used to conceptually model cause and effect relationships. Both the concept and intention domains can be linked to many other relevant GEM domains.

DISTRIBUTION LIST

A Review of Architecture Tools for the Australian Defence Force

Paul Prekop, Gina Kingston, Moira Chin, and Anna McCarthy

AUSTRALIA

DEFENCE ORGANISATION

Task Sponsor
   Chief Knowledge Officer, 1 copy

S&T Program
   Chief Defence Scientist ⎫
   FAS Science Policy ⎬ shared copy
   AS Science Corporate Management ⎪
   Director General Science Policy Development ⎭
   Counsellor Defence Science, London (Doc Data Sheet)
   Counsellor Defence Science, Washington (Doc Data Sheet)
   Scientific Adviser to MRDC Thailand (Doc Data Sheet)
   Scientific Adviser Policy and Command, 1 copy
   Navy Scientific Adviser (Doc Data Sheet and distribution list only)
   Scientific Adviser - Army (Doc Data Sheet and distribution list only)
   Air Force Scientific Adviser, 1 copy
   Director Trials, 1 copy

   Electronics and Surveillance Research Laboratory
   Director (Doc Data Sheet and distribution list only)
   Chief of ITD Division, 1 copy
   Research Leader, Joint Systems Branch, 1 copy
   Head Military Systems Synthesis Group, 1 copy
   Head Systems of Systems Group, 1 copy
   Paul Prekop, JSB, 2 copies
   Gina Kingston, JSB, 1 copy
   Moira Chin, JSB, 1 copy
   Anna McCarthy, JSB, 1 copy
   Matthew Britton, ITD, 1 copy
   Yi Yue, LOD, 1 copy
   Brendan Kirby, LOD, 1 copy
   Thea Clark, JSB, 1 copy
   Martin Burke, JSB, 1 copy

DSTO Library and Archives

Library Fishermans Bend (Doc Data Sheet)
Library Maribyrnong (Doc Data Sheet)
Library Salisbury, 1 copy
Australian Archives, 1 copy
Library, MOD, Pyrmont (Doc Data sheet only)
US Defense Technical Information Center, 2 copies
UK Defence Research Information Centre, 2 copies
Canada Defence Scientific Information Service, 1 copy
NZ Defence Information Centre, 1 copy
National Library of Australia, 1 copy

Capability Systems Staff

Director General Maritime Development (Doc Data Sheet only)
Director General Aerospace Development (Doc Data Sheet only)

Knowledge Staff

Director General Command, Control, Communications and Computers (DGC4)
(Doc Data Sheet only)
Director General Intelligence, Surveillance, Reconnaissance, and Electronic Warfare
(DGISREW)R1-3-A142 CANBERRA ACT 2600 (Doc Data Sheet only)
Director General Defence Knowledge Improvement Team (DGDKNIT)
R1-5-A165, CANBERRA ACT 2600, 1 copy
Director Defence Information Environment Architectures Office (DDIEAO) R1-3-
A138A, CANBERRA, ACT, 2600, 4 copies
Deputy Director Intelligence Development (DDID) R1-3-A032, CANBERRA ACT
2600, 1 copy
Acting Director Communications and Computer Networks (A/DCCN) R1-3-A003,
CANBERRA ACT 2600, 1 copy

Army

Stuart Schnaars, ABCA Standardisation Officer, Tobruck Barracks, Puckapunyal,
3662, 4 copies
SO (Science), Deployable Joint Force Headquarters (DJFHQ) (L), MILPO Gallipoli
Barracks, Enoggera QLD 4052 (Doc Data Sheet only)

Intelligence Program

DGSTA Defence Intelligence Organisation, 1 copy
Manager, Information Centre, Defence Intelligence Organisation, 1 copy

Corporate Support Program

Library Manager, DLS-Canberra, 1 copy

UNIVERSITIES AND COLLEGES
Australian Defence Force Academy
Library, 1 copy
Head of Aerospace and Mechanical Engineering, 1 copy
Serials Section (M list), Deakin University Library, Geelong, 3217, 1 copy
Hargrave Library, Monash University (Doc Data Sheet only)
Librarian, Flinders University, 1 copy

OTHER ORGANISATIONS
NASA (Canberra), 1 copy
AusInfo, 1 copy
State Library of South Australia, 1 copy
Parliamentary Library, South Australia, 1 copy
Codarra Advanced Systems, Fern Hill Park, Bruce, ACT 2617, 1 copy
David Eddey, Project Manager/Senior Consultant, CMG Admiral 26-28 Napier
Close Deakin ACT 2600, 1 copy

OUTSIDE AUSTRALIA

ABSTRACTING AND INFORMATION ORGANISATIONS
Library, Chemical Abstracts Reference Service, 1 copy
Engineering Societies Library, US, 1 copy
Materials Information, Cambridge Scientific Abstracts, US, 1 copy
Documents Librarian, The Center for Research Libraries, US, 1 copy

INFORMATION EXCHANGE AGREEMENT PARTNERS
Acquisitions Unit, Science Reference and Information Service, UK, 1 copy
Library - Exchange Desk, National Institute of Standards and Technology, US, 1
copy

SPARES (5 copies)

Total number of copies: **65**

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA	1. PRIVACY MARKING/CAVEAT (OF DOCUMENT) U

2. TITLE A Review of Architecture Tools for the Australian Defence Force	3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION) Document (U) Title (U) Abstract (U)

4. AUTHOR(S) Paul Prekop, Gina Kingston, Moira Chin and Anna McCarthy	5. CORPORATE AUTHOR Electronics and Surveillance Research Laboratory PO Box 1500 Salisbury SA 5108 Australia

6a. DSTO NUMBER DSTO-TR-1139	6b. AR NUMBER AR-011-848	6c. TYPE OF REPORT Technical Report	7. DOCUMENT DATE June 2001

8. FILE NUMBER N8316/22/5	9. TASK NUMBER JNT 99/018	10. TASK SPONSOR Chief Knowledge Officer	11. NO. OF PAGES 62	12. NO. OF REFERENCES 10

13. URL ON WORLDWIDE WEB http://www.dsto.defence.gov.au/corporate/reports/DSTO-TR-1139.pdf	14. RELEASE AUTHORITY Chief, Information Technology Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, SALISBURY, SA 5108

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CASUAL ANNOUNCEMENT	Yes

18. DEFTEST DESCRIPTORS

Software Tools

19. ABSTRACT
Complex defence architecture development efforts require the support of sophisticated enterprise architecture tools. This report identifies over 20 different enterprise architecture tools, and reviews four representative tools in detail. Several alternative approaches are described, including the current CAT development effort.

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE & TECHNOLOGY ORGANISATION

DSTO

ELECTRONICS AND SURVEILLANCE RESEARCH LABORATORY
PO BOX 1500 SALISBURY SOUTH AUSTRALIA, 5108
AUSTRALIA, TELEPHONE (08) 8259 5555